Properties of Gibbs samplers for inference in genetic mark-recapture models

Paula Bran

a thesis submitted for the degree of

Doctor of Philosophy

at the University of Otago, Dunedin,

New Zealand.

November 7, 2018

Abstract

The aim of this thesis is to study the convergence properties of specific MCMC algorithms for sampling from a posterior distribution. The model considered incorporates the uncertainty in the assignment of a legitimate identity of individuals. In a collected sample, observations wrongly recorded might result in duplicates or missing data which seriously affects the posterior inferences of parameters of interest. For instance, the actual sample size may be overestimated by duplicates or underestimated by the missing data. Thus, the underlying problem is a misidentification problem.

This thesis examines four MCMC algorithms. Two of which exist in the current literature (GENUAD and SMERED), however, their convergence properties had not previously been studied. This is the first contribution to the thesis. The GENUAD algorithm is a Gibbs sampler whereby the relevant full conditional densities are a critical aspect for determining the existence of a unique invariant distribution. SMERED is a Metropolis algorithm, in which convergence problems were detected. To correct these convergence issues, a novel algorithm was developed, named SMERED$^+$. Finally, the DIU algorithm attempts to propose an altogether new technique.

The comparison of the algorithms is performed by simulating the posterior distribution of interest which contains a corruption model including the uncertainty in the data. Three different datasets are considered, a fictional toy example and two collected datasets. The advantage of the toy example is the size of the state space, which allows the behaviour of the chains generated by the relevant algorithms to be observed. The other two datasets require a different treatment.

Acknowledgements

I would like to express my gratitude to my supervisors, Prof. Richard Barker and Dr. Matthew Schofield, for their patient guidance, and useful critiques during the planning and development of this research work. I appreciate their willingness to give their time generously. I am grateful to the University of Otago Doctoral Scholarship for the generous financial support. I would like to express my gratitude to Nick Gelling, a former student at the university, for his valuable help with the programming tasks. Thank you to the amazing staff members of the Department of Mathematics and Statistics who have helped me in so many different ways. I gratefully acknowledge the support staff team, Leanne, Marguerite, Greg, Chris, and Lenette (retired), and the academic staff, especially Prof. Robert Aldred, Dr. Matthew Parry, Dr. David Fletcher, Prof. Jörg Frauendiener, and Dr. Florian Beyer.

My research would have been impossible without the support of the University of Valle (Cali, Colombia), by giving me the time and financial support for my doctoral studies. I give my gratitude and appreciation to the administrative and academic staff that made this achievement possible. I am especially thankful to the Department of Mathematics for supporting the career development of the academic staff members.

I am profoundly grateful and indebted to my friend Dr. Gabrielle Knafler for her invaluable support and help while I was writing this thesis. Gabby accompanied me on this journey of writing in academic English, even when we were oceans away. I cannot imagine what this experience would have been like if I did not have her with me throughout this time. I also want to express my sincere gratitude to the other extraordinary people that I met in Dunedin, for their friendship and encouragement. I am greatly indebted to my friends that were looking out for me in the most challenging moments, their generosity and kindness gave me endurance - the right people at the right time for the right reasons. I would like to extend special thanks to some very special people that I met in a different context, Jess, Petra, and Alice at Les Mills in Dunedin. Their positive attitude and contagious enthusiasm helped me to maintain my sanity and happiness through the medium of dance and exercise.

Heart-felt thanks go to my family. The encouragement, support, and trust from my parents, brother, and sister have inspired me to embrace hope, confidence, and joy. The unconditional support from my partner, Dr. Leon Escobar who I admire profoundly, gives me the strength and determination to pursue and finish this thesis. All of you are my lighthouse in a stormy sea, shining constant and true.

"*As we express our gratitude, we must never forget that the highest appreciation is not to utter words, but to live by them*" - John F. Kennedy.

To my family, my gratitude in Spanish:

Gracias a mis viejos, Francisco y Gabriela, por apoyarme tanto, por darme todo en la vida y por esperar pacientemente todo este tiempo. Por ustedes es que estoy aquí, terminando esta tesis. Los admiro y los quiero mucho. Todo este tiempo alejados, no pasó un solo día en que no pensara en ustedes. Yuli y Cristian gracias infinitas por cuidar tan dedicamente de los viejos mientras estuve fuera. Mi Oncito, gracias amor por estar siempre conmigo, por apoyarme tanto, por creer en mí. No ha sido fácil estar separados, pero pronto llegará el día en que podamos encontrarnos en Colombia. Quiero dar especiales agradecimientos a nuestros angelitos en el cielo, nuestras abuelitas que ya no están entre nosotros. Nos vieron partir de Colombia persiguiendo un sueño, pero no tuvimos la oportunidad de celebrar con ellas la conclusión de este sueño. Mis mamitas Ana Josefa Bran y Ana María Cardona; y las abuelitas de Leon, la abuela Soledad y la mamita Estela (Q.E.D.P.). Estar en casa sin ustedes sera muy diferente. Agradecimientos a mis amigos en Colombia, que me apoyaban desde la distancia. Son muchos los nombres que tengo en mente, pero en especial quiero agradecer a Vicky, mi colega y amiga, por querer tanto a mi familia, y a Sandra por ser mi amiga leal desde la infancia.

Solo me queda decir que esta tesis no es solo mía, es de todos nosotros. Gracias por brindarme su apoyo, y sobre todo, por su AMOR.

Contents

List of algorithms

Part I

Background

Chapter 1

Introduction

1.1 Overview

Ensuring data accuracy is a question that commonly confronts statisticians. Often, the corruption sources that may contaminate the data are not considered. There is always the risk that the data contain errors which are imperceptible and unavoidable. More specifically, the observations may be duplicated, wrongly reported, or missing. For example, when conducting a laboratory experiment, the data may be corrupted due to environmental conditions and equipment failures, among other factors that are beyond the researcher's control. But not only studies utilising laboratory data produce contaminated observations. For example, a survey may contain false information of the participants due to unintentional misspelling. Other reasons for false survey data may include respondent mistakes, either intentional or unintentional, lack of interviewer impartiality, and inconsistencies with the questionnaire design.

If the integrity of the data is compromised, the misidentification of individuals (or experimental units) involved in the survey (or experiment) results in a critical problem. For example, in clinical studies, misplaced patient information can cause duplicated or incorrectly merged records which can distort the clinicians' reports. Therefore, the errors masking the true identities need to be modelled to enhance data quality, since errors can considerably affect inference. This dissertation endeavours to alleviate misidentification problems, in particular, those addressed by Wright et al. (2009) and by Steorts et al. (2016).

Wright et al. (2009) presented a model that allows the estimation of animal abundance using genetic tagging. The DNA extracted from droppings of a nocturnal animal was used as a tag in the closed-population mark-recapture study. The challenge with these natural tags is the uncertainty in the assignment of genotypes to individuals. The genotyping error was incorporated into a Bayesian model for estimating unknown quantities such as the population size and the actual sample size. A Gibbs algorithm was designed to simulate Markov chains for sampling from the relevant posterior distribution.

More generally, Steorts et al. (2016) addressed a record linkage and de-duplication problem. Often, multiple databases are combined to create a single large dataset. If the databases contain information of overlapping sets of individuals, the data may contain duplicates. Steorts et al. provided a Bayesian model for linking records across different databases that correspond to the same individual and for detecting duplicate records within them. The simulations were carried out implementing a Metropolis algorithm.

Both Wright et al. and Steorts et al. implemented Markov chain Monte Carlo (MCMC) methods for sampling from the corresponding posterior distribution in their models. They designed algorithms whose convergence properties have not been studied. In particular, the irreducibility of the Markov chain generated in Wright et al. (2009) is a critical issue to be determined, which leads to more convergence properties. Further, the proposal distribution defined in Steorts et al. (2016) cast doubts about the reversibility of the chain. Assessing the convergence of the chains is important because it could shed light on whether the posterior samples come from the respective target distributions. Thus, to determine whether these samplers converge to the invariant distribution is one of the two important contributions that this thesis offers. The second is the proposal of two novel approaches for sampling from the posterior distribution of the model developed in Wright et al. (2009).

This thesis is divided into two parts. Part I (Chapters 1-2) presents a review of the theory that is necessary to understand the content of the thesis. The remainder of this first chapter is an overview of the specific problem. Chapter 2 explains the relevant theory regarding MCMC methods because they constitute a significant component of this thesis. Part II (Chapters 3-6) characterises the original work presented in this thesis, which is the development and detailed examination of two algorithms associated with models to be used for solving misidentification problems. Although Chapter 3 is not fully an original contribution of this thesis, it presents the models and algorithms in Wright et al. (2009) and Steorts et al. (2016) addressing some inaccuracies that were found. Chapter 4 studies the theoretical convergence of the algorithms of Wright et al. and Steorts et al., presented in Chapter 3. Chapter 5 describes two new approaches for solving misidentification problems. Chapter 6 presents a comparison of the algorithms by using the badger genotypes in Wright et al. (2009). Chapter 7 discusses the results and limitations of the approaches. Finally, Chapter 8 summarises the findings and describes future research.

This thesis endeavours to solve a fundamental issue regarding the correct identification of individuals by exploring Bayesian models and MCMC algorithms. The motivating example considers the genetic data as in Wright et al. (2009). However, the primary theme of this thesis is not founded on genetics, and so details regarding genetic concepts may seem scant to some investigators. For this reason, the readers are encouraged to see Chapter 2 of Wright (2011). This chapter includes background information regarding DNA, the PCR process, microsatellite markers, and genotyping error types with approaches for managing them.

1.2 Capturing without capturing

Wildlife studies can have a negative impact on the organisms of interest, even those carried out under strict procedures and regulations. For example, Ditmer et al. (2015) found that bears experienced stress due to the presence of unmanned aerial vehicles (better known as drones) in their environment. The authors determined that bear behaviour changed drastically in the presence of these devices. In particular, they detected high heart rates, including those in hibernation. Although drones make it easier to access the natural environment of species for collecting data, they are a source of distress and disturbance. Good practices for collecting data are not the core of this dissertation; however, the citation is a good example to illustrate the stress that some species may experiment when monitored by researchers.

Non-invasive DNA samples are those obtained without invading the bodies of individuals (humans or animals) with instruments, which may cause pain or discomfort. Moreover, the presence of the individuals is not required. Examples of non-invasive DNA samples are discarded hair, skin, body fluids (such as sweat, urine), and body waste. These DNA samples have been widely used in health sciences for the prenatal assessment of some pathologies such as Down's syndrome Chiu et al. (2011) and cancer Forshew et al. (2012). In population ecology, they are often used to study elusive animals. For example, fearful wolves, shy birds, nocturnal animals or camouflaged reptiles which are virtually undetectable by eyesight.

Mark-recapture methods are often used to estimate animal abundance, which is a common problem in wildlife management. First, a sample of the population is captured, marked, and released. After some time, another sample is taken. This second sample is a mix of marked and unmarked animals. Those, which appear for the first time in the study, are marked and released along with those that were already marked. This process can be repeated as many times the researcher wants, but at least two samples are needed. The marks should allow identification of the mark-recapture history of each animal. Otis et al. (1978) described mark-recapture modelling for populations that are demographically closed, that is, no individuals enter or leave the population during the study. These models assume that marks are preserved during the experiment, meaning that they do not fall off or change in a way that they could be misread, and all marks are accurately observed and registered at each trapping occasion.

Since some mark-recapture studies have difficulties with the detection and sighting of animal populations, the concept of a "mark" continues to evolve. Marks were typically physically intrusive and constituted tags, bands, paint, and traps or nets for capturing dangerous or elusive animals. Non-invasive "marks" are potentially revolutionary because they eliminate the need to capture the animals, which may be virtually impossible depending on the species. Besides, it discharges the possibility of injuring and stressing animals during their handling. Nevertheless, the assumption regarding the accuracy of the observed marks is inappropriate because of the inherent nature of the non-invasive sampling.

Because many species avoid human contact, are nocturnal, or live in sites with restricted or difficult access for humans, alternative approaches to the traditional mark-recapture method are necessary. For example, Wright et al. (2009) developed a Bayesian model for estimating the population size of a nocturnal animal using faeces samples. Faeces are useful when the animal population has a low density, or when it is dangerous to monitor such as the large felines considered in Mondol et al. (2009) and Roques et al. (2014), and the wild canines in Marucco et al. (2012) and Morin et al. (2016). Even though marine otters are not dangerous to humans, sightings and counts are challenging, which explains the use of faeces samples by Biffi and Williams (2017). These are some examples of mark-recapture studies which use faecal DNA as natural marks.

Ecologists have taken advantage of the latest advances in molecular biology to obtain individual genotypes from non-invasive samples. The genotyped profiles of the individuals may then be used as marks because, in large populations, it is unlikely that two individuals will have the same genetic profile. The fact these samples are taken unobtrusively affects the reliability of the assignments of the genotypes to the individuals. The genotyped individuals may be subject to a high degree of uncertainty because the quality of the genetic information may be negatively affected by environmental factors or during DNA amplification. Because the use of non-invasive DNA data may be subject to genotyping error, the models in Otis et al. (1978) cannot be applied as the assumption that the marks are read and recorded correctly is inadequate.

There are some difficulties inherent in the mark-recapture approach based on DNA samples. Lukacs and Burnham (2005b) express two concerns in these studies. First, the notion of a sampling occasion is unclear. Second, it may be virtually impossible to set out a list of marks in the population. Naturally, there is concern about these difficulties because sampling occasions and marks are dominant notions in mark-recapture studies. Both issues will be discussed separately.

First, a sampling occasion refers to the time that samples are collected from the population. This concept in a conventional mark-recapture study is considered "as a short, discrete event" as stated by Lukacs and Burnham (2005b). However, in a mark-recapture study based on non-invasive DNA samples, it is vague notion. Evidence of this is the fact that the animal in an unknown time shed the DNA in the sample. Barker et al. (2014) described a general model for capture–recapture modelling of samples drawn one at a time in continuous-time. A novel aspect they included in the model is that the sampling times may be unavailable.

Second, in a standard mark-recapture study the researcher knows the list of marks in the population (for example, coloured paint, numbered tags, etc.). In DNA-based mark-recapture studies, this list is unknown because the genotypic mark is inherent to the individual. As an example, suppose that animals are marked using paint that does not wash off. Consider an initial sample of animals which were individually marked using red paint. If the second sample contains a marked animal that was not previously recorded in the first sample, then the availability of a list of marks allows the

researchers to conclude with certainty[1] that the animal has been captured for the first time in the second sample. In mark-recapture studies using non-invasive DNA, it is difficult to know whether a previously unrecorded mark (genotype) is an error in the genotyping or a new individual, unless all the genotypes in the population are known, which is virtually impossible.

Lukacs and Burnham (2005b) established that because it is impossible to know the genetic identities of every individual in the population two problems can result. First, the misidentification of individuals can occur which is better known as *genotyping error*. In traditional studies of mark-recapture, if a mark does not coincide or match with a mark from the known list, the observation is eliminated or, otherwise, corrected by the researcher. In DNA-based mark-recapture, if an incorrect genotype is logged, it is recorded as a new individual in the population. As a consequence, the size of the population will most likely be overestimated. Second, the authors point out that the marks may not be unique. In small and inbred populations, some animals may have the same genotypic profile. In this case, it is impossible to know if samples with identical genotypes are the same animal or close relatives. Consequently, the exclusion of individuals may underestimate the population.

Some models for estimating abundance incorporate the genotyping error. Lukacs and Burnham (2005a) extended the likelihood model of Otis et al. (1978) by considering the case of misidentification of individuals. They incorporated into the model the probability that a genotype (observed for the first time) is identified correctly for estimating the size of a closed population. Yoshizaki et al. (2011) further developed this model to improve the bias and precision of estimators. Wright et al. (2009) modelled the uncertainty in the assignment of genotypes to faecal pellets of badgers to estimate abundance of this species. They considered a failure produced during the process of DNA amplification which causes a genotyping error. The next section describes this error in the identification of individuals.

1.3 The genotyping error

A *gene* is a sequence of DNA that codes for a heritable trait. Genes occur at specific positions on chromosomes, called *loci*. Humans and many other organisms are diploid, meaning that they inherit one set of chromosomes from each parent. Thus, for every gene, there are two DNA sequences called *alleles*. When two alleles have the same DNA sequence, they are *homozygous*. Otherwise, they are *heterozygous*. An individual's genotype constitutes allelic combinations at loci of interest.

Polymerase chain reaction (PCR) is a technique widely used to amplify specific regions of DNA. It is relevant because researchers often want to amplify small amounts of DNA collected from the field. A common error during PCR is *allelic dropout* which

[1] The marks can be misread in standard capture-recapture sampling, and they can also fall out (e.g. the paint might wash off). If undetected, this leads to similar problems as in DNA-based studies.

means that one allele is preferentially amplified over the other, thus erroneously genotyping the sample. For a heterozygous genotype, allelic dropout can produce a false homozygote, but this failure does not occur for homozygous genotypes. For example, if an individual has a true heterozygous genotype AB at a particular locus, but the PCR amplification is only successful for allele A, then the individual will be incorrectly genotyped as an AA homozygote. Figure 1.1 shows how the true genotypes may be observed and recorded when allelic dropout is present.

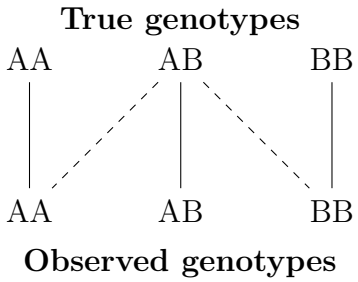


Figure 1.1: Dashed lines indicate allelic dropout. The true heterozygote AB is erroneously genotyped as AA or BB.

In large populations, allelic profiles should be unique for the sampled individuals (considering that allelic profiles consist of numerous genotyped loci). However, given the procedures and conditions for amplifying DNA, genotyping errors can be introduced which may artificially increase or decrease the variation in the population and confound individual genotypes. In particular, as shown above, the use of PCR to obtain genotypes from non-invasive DNA samples complicates the identification of individuals, because the latent (actual) identities must be determined while taking into account the uncertainty of the genetic assignments.

1.4 Data and model

The sample consists of $S = 47$ droppings of badgers collected from latrines in Woodchester Park, Gloucestershire, England. The DNA extracted from each should help to determine the identity of the individuals present in the sample. Appendix B in Wright (2011) has the information about the badger microsatellite sequences used to create the dataset, and Appendix D contains the badger data for two PCR replicates. Frantz et al. (2003) provides the original source of the dataset.

A set of $L = 7$ microsatellite marker loci was considered. They were Mel102, Mel105, Mel106, Mel109, Mel111, Mel113, and Mel117 (the abbreviation Mel refers to the scientific name for the Eurasian badger, *Meles meles*). For example, 199/199 at locus Mel105 means that both the mother and the father had a common allele and so the offspring inherited the same allele from both of the parents (i.e. homozygote at that specific locus). Alternatively, 138/142 at Mel102 means that the offspring inherited one sequence from the mother which was different from the sequence from the father

(i.e. heterozygote at that specific locus). The numbers in the genotypes indicate the sizes of the alleles (in base pairs). So, at Mel102, one sequence is 138 base pairs long while the other sequence is 142 base pairs long.

The raw numbers in a pair of microsatellite alleles is not important, but the difference between the two numbers indicates how many mutations there are between the two alleles. So, at Mel102, allele 138 has four fewer base pairs than allele 142. In medical sciences, this difference may be important for researchers looking at the association between a microsatellite sequence and a particular disease. However, in population genetics, the raw numbers and the differences between them are not directly relevant. They are used to determine whether individuals are homozygous or heterozygous at specific loci. Thus, because this thesis uses population genetic data to present a misidentification problem, the genotypes are represented as a pair of positive integers (see Definition A.1.1). If the numbers are equal, then the genotype is homozygous. Otherwise, it is heterozygous (see Definition A.1.2).

Frantz et al. (2003) used replication to overcome genotyping error. They replicated until either two alleles were detected or until they were confident of observing a homozygote. Replicate genotyping indicates the presence of allelic dropout when one replicate sample displays a heterozygote, and the other replicate sample displays a homozygote at the same locus. Under the presence of allelic dropout, there is no guarantee that the observed genotypes in the sample will allow the correct identification of the individuals.

The data is denoted by g^{obs} which comprises a $S \times L \times R$ ragged array, where g^{obs}_{jlr} is the observed genotype in the jth sample, at locus l and the rth replicate PCR amplification with $j = 1, 2, \ldots, S$, $l = 1, 2, \ldots, L$, and $r = 1, 2, \ldots, R$. The consensus genotype of an individual is an array of L pairs of alleles, since for every locus there are two alleles.

Two matrices include the latent information of the genotypes in the population and the presence of individuals in the sample, namely, $\mathcal{G}$ and X. If N denotes the number of individuals in the population, then the true genotypes in the population are arranged as the rows of $\mathcal{G}$ with order $N \times L$. The matrix X is an indicator of the presence/absence of the individuals in the sample whose dimension is $N \times S$. Each column of X corresponds to an observation in the sample and has a single 1, which indicates the association with a unique individual in the population. More formally,

$\mathcal{G} = (\mathcal{G}_{ij})$ where $\mathcal{G}_{ij}$ denotes the genotype of the ith individual in the population at the jth locus, for $i = 1, \ldots, N$ and $j = 1, \ldots, L$.
$X = (X_{ij})$ where $X_{ij} = 1$ if g^{obs}_j is realized from $\mathcal{G}_i$, and 0 otherwise.

The matrices $\mathcal{G}$ and X together constitute the latent information about which individual was caught in each sample because X is a matrix of indicators for the presence of individuals (rows) in samples (columns), and $\mathcal{G}$ contains the true genotypes (rows) which serve as unique identifiers for individuals. Other parameters included in the model are the allele frequencies γ and the dropout probability p.

The unnormalized posterior density of interest in Wright et al. (2009) is given by

$$\pi(\mathcal{G}, X, N, \gamma, p|g^{\text{obs}}) \propto \underbrace{f(g^{\text{obs}}|\mathcal{G}, X, p)}_{\text{likelihood function}} \cdot \underbrace{f(\mathcal{G}|N, \gamma) \cdot f(X|N) \cdot f(N, \gamma, p)}_{\text{prior distribution}} \tag{1.1}$$

which describes a Bayesian model for estimating the unknown parameters $\mathcal{G}, X, N, \gamma$ and p, given the observed genotypes g^{obs}. The first factor on the right side corresponds to the likelihood function, and it accounts for the corruption process contained in the data. It is defined in Appendix A.2.1. The second part includes prior distributions which will be discussed later in detail.

Simulations via MCMC methods were performed for sampling from this distribution. The supplementary material of Wright et al. (2009) explains how a Gibbs algorithm alternately updates the unknown parameters. In short, for fixed values of the other parameters, the dropout probability vector p is updated using a beta distribution; and γ using a Dirichlet distribution. The algorithm implements reversible jump MCMC for updating the population size N. The full conditional densities for updating $\mathcal{G}$ and X are categorical distributions, which are discussed later in Section 3.2. For convenience throughout this dissertation, this algorithm is called GENUAD (GENotype Uncertainty by Allelic Dropout).

Wright et al. (2009) discussed and justified the following five assumptions in the model. Section 3.2 presents more details of the model, including an example, and the algorithm used for simulating the posterior distribution in Eq. (1.1).

1. Allelic dropout is the only source of corruption for the data.
2. The probability of allelic dropout varies by loci but not individuals.
3. Allelic dropout is independent among loci and individuals.
4. The number of alleles at each locus is known.
5. Genotypes are independent among individuals.

The main focus in this thesis is how the unknown quantities $\mathcal{G}$ and X are updated. For fixed values of N, γ and p, the posterior distribution in Eq. (1.1) becomes,

$$\pi(\mathcal{G}, X|g^{\text{obs}}, N, \gamma, p) \propto f(g^{\text{obs}}|\mathcal{G}, X, p) \cdot f(\mathcal{G}|N, \gamma) \cdot f(X|N), \tag{1.2}$$

which is the posterior distribution of interest. The GENUAD algorithm, as a Gibbs sampler, simulates from this density by using the relevant full conditional densities.

Figure 1.2 illustrates the misidentification problem described above. The white nodes represent the observed genotypes g^{obs} from five DNA samples $s_1, \ldots, s_5$. The coloured nodes represent two individuals in the population, I_1 and I_2. The edges

indicate the connection between the observed genotypes and the individuals in the population. The figure shows two possible configurations of the same set of observed genotypes. In both cases, two distinct individuals were captured, but the connections are different. The configuration in Figure 1.2(a) suggests that samples s_1 and s_2 belong to the same individual I_1. Figure 1.2(b) indicates they are different individuals, with s_1 associated with individual I_2, and s_2 with individual I_1. Following the notation introduced above, the coloured nodes are elements in the matrix $\mathcal{G}$ of true genotypes, and the edges play the role of the indicator matrix X. The GENUAD algorithm aims to simulate values of $\mathcal{G}$ and X since they are unknown.

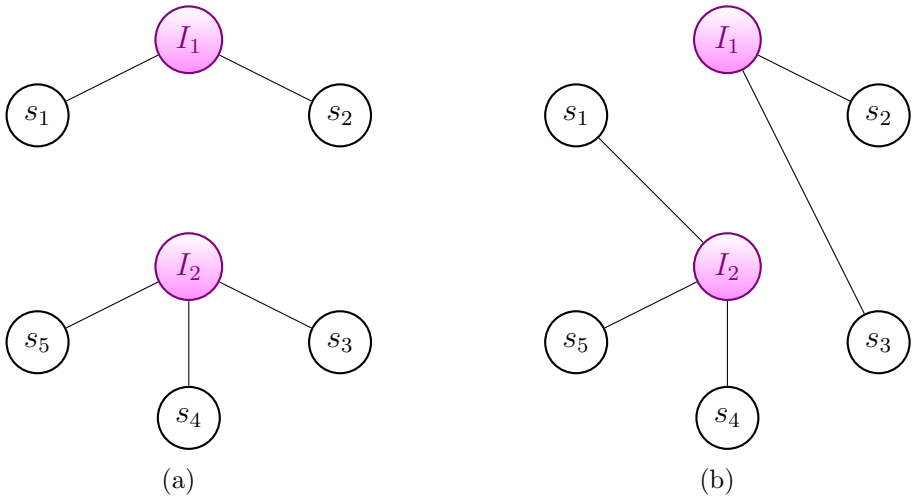


Figure 1.2: Two different configurations for five observed genotypes $s_1, \ldots, s_5$ associated with two different individuals I_1 and I_2.

The next section introduces an alternative approach which, when correctly designed, may be useful for solving the misidentification problem depicted in Figure 1.2 by sampling from the posterior distribution in Eq. (1.2).

1.5 Record linkage: An alternative approach

The consolidation of multiple sources can lead to errors. For example, the lack of unique identification numbers for individuals and non-standardized formats for the data not only lead to duplicates but also to confounding two distinct individuals and reporting them as the same. *Record linkage* connects records from two or more files that belong to the same individual.

A situation that illustrates the use of record linkage is when there are several sources providing information from individuals. For example, suppose that a university is interested in investigating the association between individual's gender, exercises habits, and the highest educational qualification. Three distinct sources may be combined to gather the required information such as the student association, the recreation services office, and the graduate school of the university. There are several drawbacks, one of

them is to determine, across the different files (sources), those students appearing in more than one file. The recognition of double-counted students avoids overestimating the population size. Thus, record linkage offers techniques for identifying the observations that refer to the same student across the three files.

Steorts et al. (2016) addressed a problem in the *record linkage* framework. The model was applied to the data from the National Long-Term Care Survey (NLTCS), which is a longitudinal study of the health of elderly (65+) individuals who stay in the survey until they either die or are lost to follow-up. The waves of the NLTCS occur at five-year intervals since 1982, and each contains approximately 20 000 individuals. Patients who die or just left the study are replaced with those who have become age 65 since the prior wave. The proposed model was used for solving simultaneously two problems: Tracking individuals across waves and detecting duplicates within waves. The hybrid MCMC algorithm developed for the simulations was named SMERED (Split and MErge REcord linkage and Deduplication). Given the similarity of the problems in Wright et al. (2009) and Steorts et al. (2016), the SMERED algorithm may suggest an alternative method for sampling from the posterior distribution in Eq. (1.2).

Record linkage has different names depending on the area of application. For example, data matching, data cleansing, coreference resolution, deduplication, duplicate detection, merge/purge, entity clustering, and householding. All refer to special cases of record linkage that are contingent upon context and purpose. Usually, statisticians and epidemiologists use the term "record linkage", while computer scientists use the term "data matching". See the seminal article Fellegi and Sunter (1969) which provided a theoretical framework for the probabilistic record linkage, widely motivated by the pioneering work of Newcombe et al. (1959). Recently, Herzog et al. (2007), Christen (2012a), Maletic and Marcus (2010) and Christen (2012b) for a modern description.

Christen (2012a) explains that the consolidation of data from different sources consists of three tasks. The first task is schema matching, which refers to the process of identifying that the meaning of words or sentences connects two objects. For example, the attribute "address" in one file could be the same attribute as "location" in another file, or maybe not. The second task is data matching (or record linkage) where records from different files that refer to the same object are linked. The third task is data fusion, which consists in the assignment of a coherent object to the records that could be referring to the same entity.

In general, record linkage methods can be deterministic or probabilistic. According to Herzog et al. (2007), deterministic record linkage is the process of linking records if they agree on a determined unique rule. A rule is a collection of identifiers called the *match key*. If the records disagree, they do not match. For example, when comparing two records which contain last names, addresses, and ages, the pair of records will be linked only if all characters in names and addresses coincide, and the numbers for the ages are exactly equal. In contrast, probabilistic record linkage involves the calculation of linkage weights which are estimated given all the observed agreements and disagreements of the records. In this case, multiple matching keys are allowed, unlike

in deterministic linkage. For example, in a clinical study, disease events may be linked to mortality data using first and last name combinations. If only a few characters are used for the first name (And* for Andrea, Andrew, Andreas), there is not a unique key for bringing two records together. Thus, when compared with deterministic linkage, the probabilistic linkage is more complicated because of errors in the linkage keys and the lack of a unique key.

Owning data also implies responsibilities such as controlling access privileges to others. The trade of data is a new trend, and it is common to see business and survey/census operations selling data. However, obtaining access to such datasets creates possible issues, such as violating the privacy of individuals or entities. For more details about the privacy protection topic, refer to Torra (2010) and Herzog et al. (2007), which examines the implications of maintaining confidentiality when record linkage is applied to improve data quality. The following paragraph from Herzog et al. (2007, p.199) speaks for itself:

> "The benefits derived from record linkage projects can be substantial both in terms of dollars saved and the timeliness of the results. The process of linking records on individuals is intrinsically privacy intrusive, in the sense that information is brought together about a person without his or her knowledge or control. So, it should not be surprising that when data are used in record linkage studies for purposes beyond which they were specifically obtained, concerns may arise regarding the privacy of those data. Privacy concerns and other legal obligations need to be met in every record linkage study."

In summary, the GENUAD and SMERED algorithms proposed by Wright et al. (2009) and Steorts et al. (2016), respectively, are suitable tools for sampling from the posterior distribution given in Eq. (1.2). Although they fit different frameworks, both solve a common problem, one where the identity of individuals in a sample needs to be determined. Figure 1.3 outlines the body of the thesis. This chapter described the misidentification problem. Chapters 2-3 include the background topics needed for understanding the original work presented in Chapters 4-7.

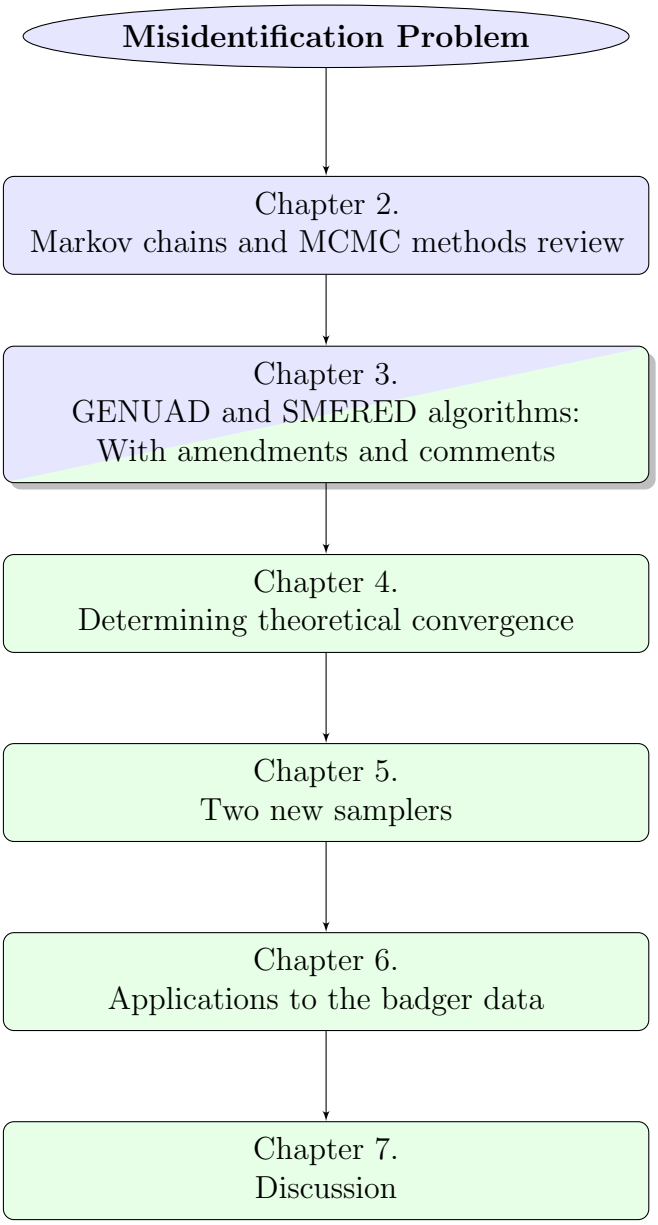


Figure 1.3: Flow chart summarising the body of the thesis. It shows the problem and the background topics (blue) needed for understanding the original work (green) in this thesis.

Chapter 2

MCMC

2.1 Basic notions of Markov chain theory

Markov chain theory forms the foundations for Markov chain Monte Carlo (MCMC) methods. It has been widely studied and applied in different areas of science (e.g. biology, health sciences, economics). There are books entirely dedicated to Markov chains such as Meyn and Tweedie (1993) that presents a meticulous study of the discrete-time Markov processes in a general state space, along with analysis of convergence. Homogeneous Markov chains in a countable state space are studied in Brémaud (1999). Levin et al. (2009) presents Markov chain theory with a strong emphasis on mixing, that is, the time to reach convergence. Robert and Casella (2004) and Athreya and Lahiri (2006) connect the study of Markov chains with MCMC simulation theory.

In this dissertation, only basic concepts and important theorems are presented. Books as Robert and Casella (2004), Brémaud (1999), Athreya and Lahiri (2006), Meyn and Tweedie (1993), Liu (1996) were very helpful for presenting this section, including the classic Cox and Miller (1965).

An important definition in Markov chain theory is the *transition kernel* as it determines the evolution of the chain. It is formally defined as follows. This definition includes some notation and elements of measure theory that are not considered in this dissertation. However, it is presented for the sake of other concepts that will be introduced later in this chapter.

Definition 2.1.1. Let $\mathcal{S}$ denote a state space and $\mathcal{B}(\mathcal{S})$ the σ-algebra of subsets of $\mathcal{S}$. A *transition kernel* is a function P defined on $\mathcal{S} \times \mathcal{B}(\mathcal{S})$ such that

i. $\forall x \in \mathcal{S}$, $P(x, \cdot)$ is a probability measure;

ii. $\forall A \in \mathcal{B}(\mathcal{S})$, $P(\cdot, A)$ is measurable.

The problem examined here is limited to discrete-time Markov chains on a discrete and finite state space $\mathcal{S}$. Brémaud (1999) defined discrete-time homogeneous Markov chains as follows.

Definition 2.1.2. Let $\{X_t\}_{t=0}^{\infty}$ be a sequence of random variables with countable state space $\mathcal{S}$. If for all integers $t \geq 0$ and all states $i_0, i_1, \ldots, i_{t-1}, i, j \in \mathcal{S}$,

$$\Pr(X_{t+1} = j | X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = \Pr(X_{t+1} = j | X_t = i), \tag{2.1}$$

provided that both sides are well-defined, the sequence is called a *Markov chain*. If the right-hand side of Eq. (2.1) is independent of t, it is called a *homogeneous* Markov chain.

Eq. (2.1) is the *Markov property*. It represents the lack of memory of the chain for remembering the states earlier than time t. Note that when $\mathcal{S}$ is discrete, the transition kernel simply is a matrix. The matrix P with entries $p_{ij} = \Pr(X_t = j | X_{t-1} = i)$ where $i, j \in \mathcal{S}$, is the *transition matrix* of the homogeneous Markov chain. In the case of discrete-time Markov chains, the transition kernel concept is replaced by transition matrix.

Notice that the transition matrix P is independent of t. That is, homogeneity refers to the fact that the law of the evolution of the chain is time-independent. The present discussion treats only homogeneous Markov chains and omits the adjective "homogeneous". Also, in later chapters, the introduction of more notation will force the use of t as a superscript rather than a subscript.

Given a transition matrix (kernel) governing the moves between states, the main goal with a Markov chain is to determine the *stationary probability distribution* π, if it exists, such that $X_t \sim \pi$ implies $X_{t+1} \sim \pi$. Ergodic chains play an important role in determining the existence and uniqueness of π. To provide this definition, some other definitions are presented first, such as irreducibility, recurrence and aperiodicity.

Communicability and irreducibility

Irreducibility is an important assumption in the theorems for determining the convergence of a Markov chain. This section introduces such a definition by following the exposition in Athreya and Lahiri (2006).

Definition 2.1.3. A state i leads to a state j ($i \to j$) if there exists a positive integer $n \geq 0$ such that $\Pr(X_n = j | X_0 = i) > 0$. A pair of states (i, j) are said to *communicate* if $i \leftrightarrow j$, that is, if there exist $n \geq 0$ and $m \geq 0$ such that $\Pr(X_n = j | X_0 = i) > 0$ and $\Pr(X_m = i | X_0 = j) > 0$.

In other words, i and j communicate if j can be reached from i in at least one step with positive probability, and vice versa. The communication relation ($\leftrightarrow$) satisfies the properties of an equivalence relation, that is, reflexivity, symmetry and transitivity.

1. Reflexivity: $i \leftrightarrow i$.
2. Symmetry: $i \leftrightarrow j$ implies that $j \leftrightarrow i$.
3. Transitivity: $i \leftrightarrow k$ and $k \leftrightarrow j$ implies that $i \leftrightarrow j$.

Reflexivity and symmetry are trivially proven. The former by choosing $n = m = 0$, $\Pr(X_0 = i | X_0 = i) = 1 > 0$. The latter holds from the commutativity in the choice of m and n. Transitivity requires further explanation. Suppose there exist positive integers n and m such that k can be reached from i in at least n steps, and j is reachable from k in at least m steps. Owing to the Chapman-Kolmogorov equations (see Robert and Casella (2004)), for reaching j from i in $n + m$ steps, an intermediate state k is necessary on the nth step. This means that a positive integer $l = n + m$ exists such that j is reachable from i in at least l steps.

Definition 2.1.4. A Markov chain with state space $\mathcal{S}$ is *irreducible* if $i \leftrightarrow j$ for every pair of states $i, j \in \mathcal{S}$.

Figure 2.1 shows an example of a reducible chain, since once state s_4 is reached, the chain is unable to move to another state.

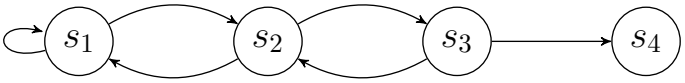


Figure 2.1: A reducible chain.

Recurrence and transience

Irreducibility refers to the ability of the Markov chain to visit every state of $\mathcal{S}$ with positive probability, starting from any other state. However, it does not give information about when the states are revisited, which can occur in a finite or infinite time. The following are concepts taken from Robert and Casella (2004) for analysing these subjects.

Definition 2.1.5. Let $\mathcal{S}$ be the discrete state space of a Markov chain. Consider $x \in \mathcal{S}$. The first time t for which the chain visits the state x, called the *stopping time* at x, is given by

$$\tau_x = \min\{t \geq 1 : X_t = x\}$$

If the chain never visits x, that is, if $X_t \neq x$ for every t, then $\tau_x = +\infty$. Additionally, the *number of passages* of the chain in x, denoted by η_x, is defined as the total of number of times that the chain enters to the state x.

The *average number of passages* in x, denoted by $\mathrm{E}(\eta_x)$, and the *probability of return* to x in a finite number of steps $\Pr(\tau_x < \infty)$ are quantities of interest because irreducibility can be either determined from the expected value or defined using the probability. Specifically, for the discrete case, Theorem 6.15 in Robert and Casella (2004) states that a chain is irreducible if and only if for every $x \in \mathcal{S}$, $\mathrm{E}(\eta_x) > 0$. Also, in the same page, the chain is defined as irreducible if $\Pr(\tau_x < \infty) > 0$ for all $x \in \mathcal{S}$.

Definition 2.1.6. In a finite state-space $\mathcal{S}$, a state $x \in \mathcal{S}$ is *recurrent* if $\mathrm{E}(\eta_x) = \infty$. If $\mathrm{E}(\eta_x) < \infty$ the state x is *transient*.

This is equivalent to saying that a state is recurrent if the chain returns to it time after time, while the transience of the state means that the chain will stop visiting it at some time. An alternative definition for recurrence is given as follows.

Definition 2.1.7. A state $x \in \mathcal{S}$ is *recurrent* if $\Pr(\tau_x < \infty) = 1$, which is equivalent to asserting that the stopping time as x is finite. The time of first return to x is called the *recurrence time*. If $\Pr(\tau_x < \infty) < 1$ then the state is *transient*.

Figure 2.2 shows recurrent and transient chains. The recurrence is clear from the fact that all the states will be revisited by the chain, while the transience is given because once the chain reaches state s_1 from s_2, the chain cannot return to s_2.

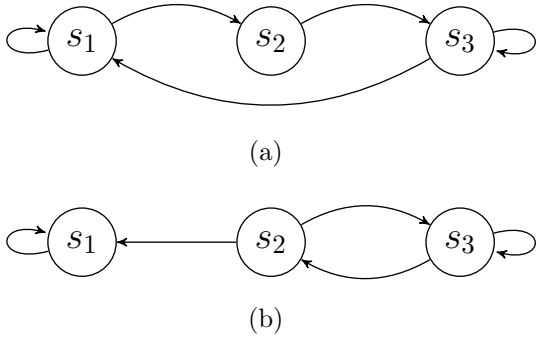


Figure 2.2: (a) A recurrent chain. (b) A transient chain.

Definition 2.1.8. A state $x \in \mathcal{S}$ is *positive-recurrent* if $\mathrm{E}(\tau_x) < \infty$. Otherwise, it is *null-recurrent*.

In other words, a state x is positive-recurrent if the mean recurrence time to x is finite which means that the chain will eventually return to x.

For the case of irreducible Markov chains, all states are recurrent or all are transient. Thus, recurrence and transience are features of the chain rather than the states. Further, based on the stability of the chains, an irreducible Markov chain can be either positive recurrent, null recurrent, or transient. In particular, Proposition 14.1.8 in Athreya and Lahiri (2006) states that for an irreducible Markov chain in a finite state space, all states are recurrent. Even more, for a finite state space irreducible Markov chain, all states are positive recurrent. This is because "the states cannot be all visited only a finite number of times; otherwise, there would exist a finite random time after which no state is visited", as Brémaud (1999) interpreted the proof of Theorem 3.3.

Definition 2.1.9. For the discrete case, a state x has *period* k if any return to state x occurs in multiples of k time steps. Formally, the period of a state $x \in \mathcal{S}$ is defined as

$$k = \text{g.c.d.}\{m \geq 1 : \Pr(X_m = x | X_0 = x) > 0\}$$

where g.c.d. is the greatest common denominator.

If two states x and y communicate, then they have the same period, which means that under irreducibility, all the states have the same period. So, for an irreducible chain, proving that for a single state its period is equal to 1 is enough for showing the aperiodicity of the chain.

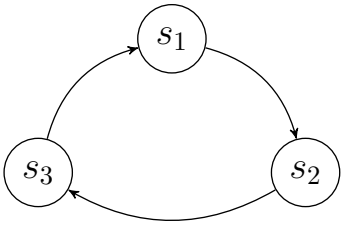


Figure 2.3: Periodic chain with period 3.

Invariant distribution

In Markov chain theory, the aim is to find the *invariant distribution* π, given a transition matrix (kernel) P from which the chain is constructed. As only discrete state spaces are treated in this thesis, then the definition of an *invariant measure* is given for this specific case, which was taken from Brémaud (1999). A definition for the continuous case is given in Robert and Casella (2004).

Definition 2.1.10. A non-null vector $\pi = (\pi_i)_{i\in\mathcal{S}}$ is called an *invariant measure* of the stochastic matrix $P = (p_{ij})_{i,j\in\mathcal{S}}$ if for all $i \in \mathcal{S}$, $\pi_i \geq 0$ and $P^T\pi = \pi$.

In this definition, $\pi_i = \pi(i)$ and P is the transition matrix. The adjective "stochastic" means that the sum of each row is equal to 1. Invariant, stationary, and equilibrium are all names that refer to the steady state of the Markov chain.

While the existence of the invariant measure is guaranteed by the positive recurrence of the chain, its uniqueness is ensured by irreducibility. These are results shown with detail in Meyn and Tweedie (1993) and Robert and Casella (2004).

Definition 2.1.11. A Markov chain is said to be *ergodic* if it is irreducible and aperiodic.

The following definition describe the most common metric for measuring how close two probability measures are. It is necessary for the convergence theorem for finite-state Markov chains which follows.

Definition 2.1.12. The *total variation distance* between two probability distributions P and Q with state space $\mathcal{S}$ is defined as

$$||P - Q||_{\text{var}} \equiv \sup_{S\subset\mathcal{S}} |P(S) - Q(S)|$$

Intuitively, for each event S in the state space $\mathcal{S}$, the variation distance between P and Q is defined as the greatest absolute difference between the probabilities assigned by both P and Q to the event S. That is, the largest possible difference between the probabilities that the distributions P and Q can assign to the same event.

The next theorem (Theorem 12.3.1 in Liu (2008)) establishes the convergence of a Markov chain in a finite state space.

Theorem 2.1.1. Consider an ergodic Markov chain with finite state space $\mathcal{S}$. Then $P^n(x_0, y) = \Pr(X_n = y|X_0 = x_0)$ as a probability measure on y converges to its invariant distribution $\pi(y)$ geometrically in variation distance. That is, there exist $0 < r < 1$ and $c > 0$ such that

$$||P^n(x, \cdot) - \pi||_{\text{var}} \leq cr^n \tag{2.2}$$

where $||\cdot||_{\text{var}}$ denotes the total variation distance explained in Definition 2.1.12.

Reversibility

This section is based on Robert and Casella (2004).

Definition 2.1.13. A stationary Markov chain (X_t) is *reversible* if the distribution of X_{t+1} conditionally on $X_{t+2} = x$ is the same as the distribution of X_{t+1} conditionally on $X_t = x$.

In other words, reversibility can be interpreted as the ability of the chain to evolve in the same way backwards and forwards.

Definition 2.1.14. Let (X_t) be a Markov chain with transition matrix P and state space $\mathcal{S}$. It is said that P satisfies the *detailed balance equation* if there exists a function f such that

$$f(x)P(x, y) = f(y)P(y, x) \tag{2.3}$$

for all $x, y \in \mathcal{S}$.

Eq. (2.3) expresses the symmetry in the evolution of the Markov chain; specifically, it says that the probability of a move to y from x is equal to the probability of the return move to x coming from y. The next result corresponds to Theorem 6.46 in Robert and Casella (2004). It is relevant because it establishes that reversibility is a sufficient condition for ensuring the existence of the stationary distribution.

Theorem 2.1.2. Suppose that a Markov chain with transition matrix P satisfies the detailed balance equation with π a probability density function. Then:

i. The density π is the invariant density of the chain.

ii. The chain is reversible.

To summarize this section, consider the case of an ergodic Markov chain with state space $\mathcal{S}$, transition kernel P, and invariant distribution π. If P satisfies the detailed balance equation, with π playing the role of function f in Eq. (2.3), then π is the unique stationary distribution of the chain. With the existence of π, the detailed balance equation and reversibility are equivalent. Thus, knowing that a Markov chain is reversible leads to the existence of a unique stationary distribution. However, it is not a necessary condition. In general, for those Markov chains that are not reversible, there is no guarantee about the existence of a stationary distribution. In those cases, proving the existence of a unique stationary distribution can be a hard task. The next example, taken from Sorensen and Gianola (2002), illustrates the situation of a non-reversible Markov chain with an invariant stationary distribution π.

Example 2.1.1. Consider a chain with state space $\mathcal{S} = \{0, 1, 2\}$ and transition matrix P given by

$$P = \begin{pmatrix} 7/10 & 2/10 & 1/10 \\ 1/10 & 7/10 & 2/10 \\ 2/10 & 1/10 & 7/10 \end{pmatrix}.$$

The non-reversibility of the chain is easily checked because $\pi(0)P(0,1) = 1/15$, which is different from $\pi(1)P(1,0) = 1/30$. For large n,

$$P^n = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}.$$

Although the chain is not reversible, the unique invariant distribution is $\pi = (1/3, 1/3, 1/3)'$ as $P^n\pi = \pi$. □

2.2 Markov chain Monte Carlo methods

Probabilistic modelling commonly requires integrating complex and multidimensional probability distributions. An example is the calculation of the expectation of the model distribution. Often, calculating these integrals is impossible because either there is no a closed-form expression for the integral or the high dimensionality of the distribution. Markov Chain Monte Carlo (MCMC) methods can be used in such situations. As the name suggests, the machinery of MCMC methods comprises two essential parts: Markov chains and Monte Carlo integration. Thus, MCMC simulation approximates complex integrals using stochastic sampling routines derived from Markov chain theory.

Monte Carlo integration proceeds by sampling the distribution in question, and then approximating the problematic integration by appropriate numerical summary and relying on the law of large numbers. However, sampling from the probability distribution may be difficult or impossible to do directly. This is when the Markov chain element takes part in the solution of the problem. From Definition 2.1.2, a Markov chain is a sequence of states of a random variable which is generated by using a transition rule such that the moves between states are probabilistic and the future state is

only conditioned on the preceding state. Under certain conditions, the Markov chain will generate states which are draws from a target probability distribution. Therefore, a MCMC algorithm generates samples from a distribution (Markov chain) whose integral needs to be approximated (Monte Carlo integration).

MCMC methods have been widely studied and discussed from a theoretical point of view [Tierney (1994), Robert and Casella (2004), Gelman et al. (2004), Lange (1999), Athreya and Lahiri (2006)], and applied in diverse areas such as molecular biology, ecology, zoology, etc. Link and Barker (2010) present several examples applied to ecology and relevantly discuss the ergodicity theorem. Sheehan and Thomas (1993) were the first to address a problem involved with irreducibility in the context of a single-genotype Gibbs sampler for genetic loci. Subsequently, Lin et al. (1993) modified the conditions to achieve irreducibility using a variation of Metropolis samplers. Likewise, Lange (2002) states that the application of Markov chain Monte Carlo methods in the analysis of human pedigree data requires a clear definition of "an appropriate state space and a mechanism for moving between neighboring states of the space". Thompson (2000) summarizes how the Monte Carlo methods are applied to genetic structures.

A variety of MCMC algorithms have been developed for constructing Markov chains with a desired invariant distribution. The most popular are Metropolis-Hastings and Gibbs algorithms. There are also hybrids of these algorithms which are proposed for improving convergence. The next sections briefly introduce these algorithms.

2.2.1 Gibbs sampler

Gibbs sampling became popular in statistical physics as a solution to an image processing problem in a paper by Geman and Geman (1984). It was named in honour of the physicist Josiah Willard Gibbs (1839-1903) who is a founder of statistical mechanics. Although this method has been studied by both statisticians and physicists, it has become a useful tool for researchers from many other disciplines wishing to model data and make inferences about a particular phenomenon. This spread of applications has been favoured by the appearance and propagation of computers, which translates in the rapid increase of the number of publications implementing the method.

The advantage of Gibbs sampling is its simplicity when applied. In contrast, the theory behind might be a little obscure, especially for researchers who are not familiarized with the theory involved by the full conditional densities used for generating the Markov chain. With some exceptions, many applied researchers (ecologists, geneticists, computer scientists, etc.) apply the method without being interested in questions such as why it works. Then, the general rule in these cases is trying to keep the theory as minimum as possible. Thus, thanks to the availability of a wide variety of software implementing Gibbs algorithms, those researchers do not need (and do not have) to go deep in theoretical questions.

An important feature of the Gibbs algorithm is that multivariate distributions can be sampled using univariate conditional densities. A careful choice of these densities

ensures the success of the simulations. There are two ways of updating the parameters, by *random* or *systematic* scan Gibbs sampling. In this section, only the latter is shown including some aspects about its convergence. Refer to Robert and Casella (2004), Casella and George (1992), Brémaud (1999) for more details on Gibbs sampling.

Definition 2.2.1. Let $\boldsymbol{X} = (X_1, \ldots, X_q)$ be a random variable where $q > 1$ and $\dim(X_i) = 1$, for $i = 1, \ldots, q$. Denoting the corresponding univariate conditional densities by $f_1, \ldots, f_q$, assume that draws from these distributions are possible, that is, for $i = 1, \ldots, q$,

$$X_i|\boldsymbol{x}_{-i} \sim f_i(x_i|\boldsymbol{x}_{-i})$$

where $\boldsymbol{x}_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_q)$.

Setting a state $\boldsymbol{X}^{(t)}$, a transition to the state $\boldsymbol{X}^{(t+1)}$ by a systematic Gibbs sampling algorithm is given by q steps:

1. $X_1^{(t+1)} \sim f_1(x_1|x_2^{(t)}, \ldots, x_q^{(t)})$
2. $X_2^{(t+1)} \sim f_2(x_2|x_1^{(t+1)}, x_3^{(t)}, \ldots, x_q^{(t)})$

$\vdots$

q. $X_q^{(t+1)} \sim f_q(x_q|x_1^{(t+1)}, \ldots, x_{q-1}^{(t+1)})$

The univariate densities $f_1, \ldots, f_q$ are called *full conditional* and they are a critical issue for discussing the convergence of a Gibbs algorithm in the next section.

Hereafter, just "Gibbs sampling" will refer to the systematic scan. The difference with the random scan Gibbs sampler, as its name indicates, is the random choice of a parameter to be updated given the rest of them. See Robert and Casella (2004), and Levine and Casella (2006).

Convergence properties of the Gibbs sampler

The strength of Gibbs sampling is the recovery of the joint density by sampling from the full conditional densities. To illustrate, consider the bivariate case for which the joint density can be expressed by using the conditional densities $f_{X|Y}$ and $f_{Y|X}$ as follows.

$$f_{X,Y}(x, y) = f_{X|Y}(x|y)f_Y(y) = f_{Y|X}(y|x)f_X(x).$$

From here,

$$\int \frac{f_{Y|X}(y|x)}{f_{X|Y}(x|y)}dy = \int \frac{f_Y(y)}{f_X(x)}dy = \frac{1}{f_X(x)}$$

Then, the joint density can be written as

$$f_{X,Y}(x, y) = \frac{f_{Y|X}(y|x)}{\int f_{Y|X}(y|x)/f_{X|Y}(x|y)dy} \tag{2.4}$$

provided that the involved quantities exist, especially the integral in the denominator. For example, consider the conditional densities $X|Y = y \sim \text{Exp}(y)$ and $Y|X = x \sim \text{Exp}(x)$. From which, $f_{Y|X}(y|x)/f_{X|Y}(x|y) = x/y$. In this case, the integral in the denominator of Eq. (2.4) diverges. Then, the joint density associated with the full conditional densities $f_{Y|X}$ and $f_{X|Y}$ does not exist. Thus, Eq. (2.4) is valid under the assumption of existence of the joint density.

The example above was taken from Hobert and Casella (1998) who generalized the concept of *compatibility* between conditional distributions. This concept had been previously introduced by Arnold and Press (1989).

Definition 2.2.2. Given a set of conditional densities $(f_1, \ldots, f_q)$, they are *compatible* if there exists a joint density f such that for each $i = 1, \ldots, q$,

$$f_i(x_i|\boldsymbol{x}_{-i}) = \frac{f(\boldsymbol{x})}{\int f(\boldsymbol{x})dx_i}$$

for all $\boldsymbol{x} = (x_1, \ldots, x_q) \in \text{supp}(f)$. If f is unique, then they are *strongly compatible*.

The support of a function f, denoted by $\text{supp}(f)$, is understood as the subset of the domain values for which the function is positive. For the special case of density functions, it is the set of elements in the sample space that have positive probability.

Arnold and Press (1989) proposed sufficient and necessary conditions for determining compatibility in the bivariate case. The authors determined the uniqueness of the invariant distribution from the irreducibility of a particular matrix, which is the product of the transition matrices in the context of Markov chains. This approach is not considered here because the transition matrices for the specific problem in this work are not available. Refer to Hobert et al. (1997), in addition to the previous references.

For the particular interest in this thesis, the Hammersley-Clifford (H-C) theorem, which will be presented as Theorem 2.2.1, is an important key for establishing the existence of the unique invariant distribution. It expresses a joint density using the corresponding full conditional densities, under the assumption that the joint density satisfies the *positivity condition* which is presented as Definition 2.2.3. See Besag (1974) and the unpublished work by Hammersley and Clifford (1971) are key references for more detail on the H-C theorem and the positivity condition.

Definition 2.2.3. Let $(X_1, \ldots, X_q)$ be random variables with joint density denoted by f. If $f_{X_i}(x_i) > 0$ for every $i = 1, \ldots, q$, where f_{X_i} denotes the marginal distribution of X_i, implies that $f(x_1, \ldots, x_q) > 0$, then f is said to satisfy the *positivity condition*.

Thus, if the support of a joint density f is the Cartesian product of the supports of the marginal densities f_{X_i}, then f satisfies the positivity condition. That is, if

$$\text{supp}(f) = \text{supp}(f_{X_1}) \times \ldots \times \text{supp}(f_{X_q}),$$

then f satisfies the positivity condition.

The Hammersley-Clifford theorem is stated as follows and the proof can be also found in Robert and Casella (2004) as Theorem 10.5.

Theorem 2.2.1 (The Hammersley-Clifford theorem)**.** Suppose a distribution whose density $f(x_1, \ldots, x_q)$ satisfies the positivity condition. Then, for any $(z_1, \ldots, z_q) \in \text{supp}(f)$,

$$f(x_1, \ldots, x_q) \propto \prod_{i=1}^{q} \frac{f_i(x_i|x_1, \ldots, x_{i-1}, z_{i+1}, \ldots, z_q)}{f_i(z_i|x_1, \ldots, x_{i-1}, z_{i+1}, \ldots, z_q)}$$

where f_i's denote the full conditional densities.

Theorem 2.2.1 establishes that a joint density, which satisfies the positivity condition in Definition 2.2.3, is uniquely determined by the full conditional densities. Nevertheless, the positivity condition can be a demanding condition that may not always be satisfied. Besag (1994) in a discussion of Tierney (1994) proposed a relaxed condition, due to the lack of positivity constraints "in some applications where there are deterministic exclusions between the values taken by the X_i's". The positivity condition requires positive marginal densities, however, Besag's condition only requires the joint density to be positive. The condition is presented as a lemma and its proof can be found in Besag (1994).

Lemma 2.2.1. Let χ be the support of f, that is $\chi = \{\boldsymbol{x} \in \mathbb{R}^n : f(\boldsymbol{x}) > 0\}$. If for each $\boldsymbol{x} \in \chi$ and a fixed initial state $\boldsymbol{x}^0 \in \chi$, there exists a finite sequence $\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^m = \boldsymbol{x}$ of states in χ such that consecutive states differ in a single component, then the full conditional densities $f_i(x_i|\boldsymbol{x}_{-i})$, $i = 1, 2 \ldots, n$, determine f.

This lemma not only establishes that the full conditional densities determine a unique joint density, under the assumption that this is positive, but also that the existence of the finite sequences ensures irreducibility. That is, all the states of the state space can be visited for the chain with positive probability.

Hobert et al. (1997) argued for the need to reinforce the condition placed on the joint density. They showed with an example that Lemma 2.2.1 could fail for a continuous state space. Their counterexample illustrates a joint density which is positive, but it is not determined by the full conditional densities. That is, the Besag condition is satisfied but ergodicity is not. Hobert et al. proposed an adjustment such that the problem of defining the conditional distributions on sets of measure zero is evaded. Then, both the uniqueness of the joint distribution and the ergodicity of the Gibbs chain are ensured.

As discussed in Section 2.1, irreducibility is a crucial property for determining the convergence of a Markov chain, and it is closely related to the positivity condition. The next theorem is presented in Robert and Casella (2004) as Lemma 9.5.

Theorem 2.2.2. For the Gibbs sampler, if the joint density satisfies the positivity condition defined in Definition 2.2.3, then the Gibbs Markov chain is irreducible.

The result follows almost directly from Theorem 2.2.1. Since it implies that, for a joint density holding the positivity condition, all its full conditional densities are positive as well. This means that sampling from any of the full conditional does not take the Markov chain into a region outside the support of the joint density. Instead, positivity means that the Markov chain generated by the Gibbs sampler can travel between regions in the state space in a single iteration with positive probability.

Theorem 2.2.2 suggests that Gibbs sampling may generate reducible Markov chains, as the next example shows, which was taken from Robert and Casella (2004).

Example 2.2.1. Let $\mathcal{E}$ and $\mathcal{E}'$ denote disks with radius 1 and centers $(1, 1)$ and $(-1, -1)$, respectively. Consider the distribution with density

$$f(x_1, x_2) = \frac{1}{2\pi}[\mathbb{1}_{\mathcal{E}}(x_1, x_2) + \mathbb{1}_{\mathcal{E}'}(x_1, x_2)]. \tag{2.5}$$

Figure 2.4 shows $\text{supp}(f)$. A Gibbs sampler is unable to generate an irreducible chain because the disks are located in different quadrants of the plane. If the chain starts in an element in $\mathcal{E}$, it is unable to move to $\mathcal{E}'$. As mentioned by the authors, a change of coordinates solves the problem of reducibility.

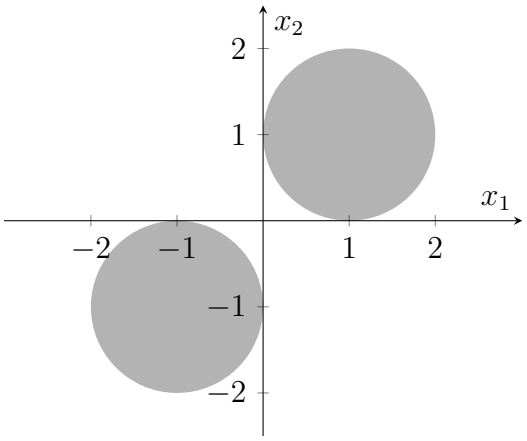


Figure 2.4: The non-connected support of the joint density f in Eq. (2.5).

Setting $z_1 = x_1 + x_2$ and $z_2 = x_1 - x_2$, the original coordinates are expressed as $x_1 = (z_1 + z_2)/2$ and $x_2 = (z_1 - z_2)/2$. Then, the joint density of z_1 and z_2 is given by

$$g(z_1, z_2) = \frac{1}{4\pi}[\mathbb{1}_{\mathcal{D}}(z_1, z_2) + \mathbb{1}_{\mathcal{D}'}(z_1, z_2)],$$

where $\mathcal{D}$ and $\mathcal{D}'$ are disks with radius $\sqrt{2}$ and centers $(2, 0)$ and $(-2, 0)$. The existence of at least one pair of coordinates differing in a single component (abscissa or ordinate) implies the connectedness of $\text{supp}(g)$.

□

The example illustrates why connectedness of the support is a crucial point when irreducibility is required. This depends on the specific problem, and in some cases, it is impossible to guarantee that the positivity condition holds for the joint density.

If $x^{(t)} = \left(x_1^{(t)}, \ldots, x_q^{(t)}\right)$ represents the state of the chain at time t, under positivity of the joint density, the transition kernel of the systematic scan Gibbs sampler, for passing from $x^{(t-1)}$ to $x^{(t)}$, is given by

$$\begin{aligned} P(x^{(t-1)}, x^{(t)}) &= f_1\left(x_1^{(t)} | x_2^{(t-1)}, \ldots, x_q^{(t-1)}\right) \cdot f_2\left(x_2^{(t)} | x_1^{(t)}, x_3^{(t-1)}, \ldots, x_q^{(t-1)}\right) \\ &\quad \ldots f_q\left(x_q^{(t)} | x_1^{(t)}, x_2^{(t)}, \ldots, x_{q-1}^{(t)}\right). \end{aligned} \tag{2.6}$$

This result is given in Robert and Casella (2004) as Theorem 10.6. Also, it is well known that a Markov chain generated by the systematic scan Gibbs sampler is not reversible, because the transition kernel P in Eq. (2.6) does not satisfy the detailed balance equation,

$$f(x^{(t-1)})P(x^{(t-1)}, x^{(t)}) \neq f(x^{(t)})P(x^{(t)}, x^{(t-1)}).$$

In contrast, the random scan Gibbs sampling is reversible. However, non-reversibility is not a problem for the performance of the algorithm. Even when reversibility is a sufficient condition for concluding the existence of an invariant distribution, it is not a necessary condition, see Theorem 2.1.2. Thus, non-reversibility does not imply the non-existence of the invariant distribution.

2.2.2 Metropolis-Hastings algorithm

The Gibbs algorithm presented in the previous section is a special case of the Metropolis-Hastings (M-H) algorithm albeit their convergence is evaluated differently. This section is based on several sources such as Chib and Greenberg (1995), Gelman et al. (2004), Robert and Casella (2004).

As discussed earlier, for MCMC methods the invariant density π (target distribution) is known while the transition kernel is unknown. For generating samples from π, a suitable transition kernel must be found. From Chib and Greenberg (1995):

> "MCMC methods turn the theory around [when compared with the construction of a Markov chain]: the invariant density is known (perhaps up to a constant multiple)—it is $\pi(\cdot)$, the target density from which samples are desired—but the transition kernel is unknown."

Denote the *candidate-generating density* (or *proposal distribution*) by $q(x, x')$. The arguments indicate that starting in state x, the density generates a state x' from $q(x, x')$. If q satisfies the detailed balance equation then it will be the required transition kernel, but for most of the cases it is not satisfied. Thus, the reversibility condition will be essential for constructing the required transition kernel. For the case that q does not satisfy the detailed balance equation condition, consider the inequality:

$$\pi(x)q(x, x') > \pi(y)q(x', x) \tag{2.7}$$

This expression says that the chain is more inclined to move from x to x' rather than in the opposite way. Chib and Greenberg (1995) explain that in order to balance the equation the number of moves from x to x' must be reduced by the addition of the probability of occurrence of that move, $\alpha(x, x') < 1$. They refer to $\alpha(x, x')$ as the *probability of move*. The chain returns to x if the move does not occur. Then,

$$P_{MH}(x, x') = q(x, x')\alpha(x, x'), \tag{2.8}$$

is defined for governing the transitions from x to x', with $x \neq x'$, such that it satisfies the reversibility condition, that is,

$$\pi(x)q(x, x')\alpha(x, x') = \pi(x')q(x', x)\alpha(x', x). \tag{2.9}$$

The probability $\alpha(x, x')$ in this equation should be defined such that Eq. (2.7) is balanced, which happens when $\alpha(x', x) \approx 1$, that is, when the probability of a move to x from x' is sufficiently large. Thus, $\alpha(x, x')$ is defined such that

$$\alpha(x, x') \approx \frac{\pi(x')q(x', x)}{\pi(x)q(x, x')}.$$

Chib and Greenberg (1995) explain that to guarantee that the transition kernel $P_{MH}(x, x')$ satisfies the reversibility condition (i.e. Eq. (2.9)), the probability of move $\alpha(x, x')$ needs to be defined as

$$\alpha(x, x') = \begin{cases} \min\left[\dfrac{\pi(x')q(x', x)}{\pi(x)q(x, x')}, 1\right] & \text{if } \pi(x)q(x, x') > 0 \\ 1 & \text{otherwise.} \end{cases}$$

Therefore, the M-H algorithm can be summarized as follows. It starts in an initial value x^0 such that $\pi(x^0|\text{data}) > 0$. For $t = 1, 2, \ldots$

1. Generate a candidate x^* from the proposal distribution $q(x^{t-1}, \cdot)$.

2. Calculate the ratio

$$r = \frac{\pi(x^*)q(x^*, x^{t-1})}{\pi(x^{t-1})q(x^{t-1}, x^*)} \tag{2.10}$$

3. Set

$$x^t = \begin{cases} x^* & \text{with probability } \min(1, r) \\ x^{t-1} & \text{otherwise.} \end{cases}$$

The set of accepted proposals generates a Markov chain that should converge to the target distribution π. When the proposal distribution is symmetric, that is, when $q(x, x') = q(x', x)$, the ratio r is expressed only in terms of the target density as $r = \pi(x^*)/\pi(x^{t-1})$. In this case, it is referred to as *Metropolis algorithm*. As stated in Gelman et al. (2004), asymmetric proposal distributions can be beneficial in order to speed up the evolution of the chain.

A difficulty with the M-H algorithm is the choice of an appropriate proposal distribution because the wrong selection can lead to several problems regarding the convergence of the Markov chain. Chib and Greenberg (1995) provided some guidance on implementation. They stated that the behaviour of the chain is influenced by the range of the proposal distribution. In particular, it affects the acceptance rate and the sample space of the chain. Further discussion about this can be found on section 5 in Chib and Greenberg (1995).

As discussed in Robert and Casella (2004), the proposal distribution q and the density π above must hold some "minimal regularity conditions" to guarantee that π will indeed be the stationary distribution of the chain. A necessary condition is presented in the next theorem.

Theorem 2.2.3. Consider the Markov chain generated by the M-H algorithm defined above. For every proposal distribution q such that $\text{supp}(\pi) \subset \text{supp}(q)$,

i. the kernel of the chain satisfies the detailed balance equation with π,

ii. π is the invariant distribution of the chain.

The proof of the theorem can be found in Robert and Casella (2004, p. 272), but an intuitive explanation could be as follows. Suppose that $\text{supp}(\pi)$ is not a subset of $\text{supp}(q)$. If a Markov chain is started in $x^{(0)} \in \text{supp}(q)$, then the chain is unable to visit the space corresponding to $\text{supp}(\pi) \setminus \text{supp}(q)$. That is, π would not be the invariant distribution. Thus, a necessary condition for π to be the invariant distribution is that $\text{supp}(\pi)$ must be a subset of $\text{supp}(q)$.

Positivity of the proposal distribution q is a sufficient condition for ensuring the irreducibility of a Markov chain generated by the M-H algorithm. That is, if $q(x, x') > 0$ for every $x, y \in \text{supp}(\pi)$, then all the states in the state space communicate. Irreducibility implies positive recurrence of the chain if the state space is finite. Aperiodicity is satisfied if the event $\{X^{(t+1)} = X^{(t)}\}$ has positive probability which is equivalent to

$$\Pr[\pi(x^{t-1})q(x^{t-1}, x^*) \leq \pi(x^*)q(x^*, x^{t-1}))] < 1.$$

2.2.3 Reversible jump MCMC

The M-H algorithm operates in spaces with fixed dimension. That is, the states x and x', as above, lie in spaces with the same dimension. Green (1995) proposed a reversible jump Markov chain Monte Carlo (RJMCMC) algorithm, which allows transitions between states with different dimension. In this sense, RJMCMC can be considered as an extension of the M-H algorithm. It has been used in different applications. A simple example is model selection procedures where different model structures are evaluated and contrasted to obtain the optimal representation of the data. The parameter space may be different for all the candidate model structures, and even more, they may have different dimensions.

Waagepetersen and Sorensen (2001) implemented the technique in genetics to compute the posterior distribution of the number, locations, effects, and genotypes of putative quantitative trait loci. Hastie and Green (2012) used this approach for model determination problems. Sisson (2005) offers a detailed review about the advances ten years after the release of the technique. Barker and Link (2013) formulated RJMCMC as Gibbs sampling with alternating updates of a categorical variable indicating the choice of a model, and a "palette" of parameters, which allows the construction of the parameters for a given model. The presentation of the technique will be presented in this section following Green (2003) and Hastie and Green (2012).

The simulation of target distributions on spaces of fixed dimension is itself a complex problem, which becomes even more challenging when the dimension is changing. Green (2003) describes this situation as "the number of things you don't know is one of the things you don't know". The author states that problems of this nature can be specified as a joint inference problem of a model indicator k and a parameter vector θ_k. The model indicator determines the dimension n_k of the parameter, which differs between models. Thus, the aim is to use the joint posterior density $f(k, \theta_k|y)$ for inference.

Suppose a prior distribution $f(\theta_k|k)$ and a likelihood $f(y|k, \theta_k)$ for the data y, given a prior $f(k)$ over model indicators k in a countable set $\mathcal{K}$, for each k. For simplicity, θ_k is assumed as n_k-dimensional and there are no other parameters. Thus, when the models coincide in some parameters, these are included in $\theta_k \in \mathbb{R}^{n_k}$ and never considered separately.

By virtue of a MCMC approach the joint posterior $\pi \equiv f(k, \theta_k|y)$ can be computed by constructing a Markov chain with state space $\mathcal{X}$ comprised of pairs (k, θ_k). That is,

$$\mathcal{X} = \bigcup_{k \in \mathcal{K}} (\{k\} \times \mathbb{R}^{n_k}). \tag{2.11}$$

RJMCMC requires multiple types of moves to traverse the entire space $\mathcal{X}$. Each move type is a transition kernel reversible with respect to π. This trans-dimensional approach extends the M-H algorithm because the probabilities of each move type depend on the current state.

The move types are indexed by m in a countable set $\mathcal{M}$. A move type $m \in \mathcal{M}$ comprises both the forwards move from $x = (k, \theta_k)$ to $x' = (k', \theta'_{k'})$, and the reverse from x' to x, for a specific pair (k, k'). For the forwards move, an auxiliary random variable u with dimension r is generated from a known joint distribution called *jumping* distribution $J_{k \to k'}$. A transformation of θ_k and u will define the pair $(\theta'_{k'}, u')$ where u' with dimension r' is generated from a joint distribution $J_{k' \to k}$ required for the reverse move. So, $(\theta'_{k'}, u') = h_m(\theta_k, u)$. The inverse function h'_m of h_m allows the reverse move.

The move-type specific transition kernel equivalent to Eq. (2.9) is

$$\pi(x) J_{k \to k'}(u) j_m(x) \alpha_m(x, x') = \pi(x') J_{k' \to k}(u') j_m(x') \alpha_m(x', x) \tag{2.12}$$

where $j_m(x)$ is the probability of move m when at state x. As before, the aim is to find the probability of moving from x to x', $\alpha_m(x, x')$ such that $J_{k\to k'}(u)j_m(x)$ satisfies the detailed balance equation. The acceptance probability α_m associated with move type m is defined as

$$\alpha_m(x, x') = \min\left\{1, \frac{\pi(x')J_{k'\to k}(u')j_m(x')}{\pi(x)J_{k\to k'}(u)j_m(x)}\left|\frac{\partial(\theta'_{k'}, u')}{\partial(\theta_k, u)}\right|\right\}. \tag{2.13}$$

To guarantee that the transformation from (θ_k, u) to $(\theta'_{k'}, u')$ is a diffeomorphism, it is necessary that $n_k + r = n_{k'} + r'$, which is known as dimension matching. The transformation $(\theta_k, u) \mapsto (\theta'_{k'}, u')$ is required to be a diffeomorphism, because it will ensure that the transformation and its inverse are differentiable. The Jacobian factor in Eq. (2.13) results of applying the change of variables technique to the jumping distribution $J_{k\to k'}$ and the transformation $(\theta_k, u) \mapsto (\theta'_{k'}, u')$.

Therefore, the remarkable feature of the RJMCMC approach is the convenient use of augmenting (auxiliary) variables u, whose role is to match dimensions, because the parameters may live in spaces of different dimensions. The following frame summarises the RJMCMC algorithm.

RJMCMC algorithm
For a specific pair (k, k'), for moving from $x = (k, \theta_k)$ to $x' = (k', \theta'_{k'})$, where $\dim(\theta_k) = n_k$ and $\dim(\theta'_{k'}) = n'_k$,

1. Choose a move m with probability $j_m(x)$.
2. Generate $u \sim J_{k\to k'}(u)$.
3. Define a one-to-one transformation h_m such that $(\theta_k, u) \mapsto (\theta'_{k'}, u')$.
4. Accept the new model k' with probability equal to $\min(1, r)$ where

$$r = \frac{\pi(x')J_{k'\to k}(u')j_m(x')}{\pi(x)J_{k\to k'}(u)j_m(x)}\left|\frac{\partial(\theta'_{k'}, u')}{\partial(\theta_k, u)}\right|. \tag{2.14}$$

The notation for explaining the RJMCMC algorithms varies from author to author. This is common when mathematical formalism is involved. However, such a wide variety of notation could create confusion for some researchers when attempting to implement the technique. This statement is supported by Hastie and Green (2012), where the reluctance to apply the reversible jump approach is attributed to the intricate and rigorous mathematical language behind. However, they highlight that it is unnecessary to master the formalities to use it for modelling. The main concern is choosing the appropriate proposal distribution because it will determine the speed of the Markov chain for exploring the state space. As the authors remark, the effect of an efficient proposal distribution for moving between states is the fast exploration of state space by the chain.

Hastie and Green (2012) characterise two main approaches using MCMC for model determination problems. *Across-model* simulation, in which the states have the form $(k, \theta_k) \sim f(k, \theta_k|y)$, is the subject in RJMCM. *Within-model* simulation, in which there are independent simulations $\theta_k \sim f(\theta_k|k, y)$ for each model k, refers to the simulations in a fixed-dimension context. Hastie and Green state that, for within-model proposals, the concept of closeness or neighbouring between states makes sense. That is, for a given model with fixed-dimension parameter space, a proposed state is accepted with high probability if it is close to the current state. But, failing to sample from areas that are far apart, or proposing states with very low acceptance probabilities could lead to an inefficient chain. In contrast, for the case of across-model proposals, the concept of closeness between states cannot be established. Accordingly, Hastie and Green asserted that the foundation for devising an efficient across-model proposal is to assure that the current state (k, θ_k), and the new state $(k', \theta'_{k'})$, will have similar posterior supports. In this way, the proposal will have a high probability of being accepted. More details about the choice of efficient proposals can be found in Hastie and Green (2012).

2.2.4 Metropolized independent sampling

The definition of the candidate-generating density, also called the proposal distribution, was seen in Section 2.2.2 as a function $q(x, y)$ which generates a state y given the current state x. There is a clear dependence of the proposal from the previous state. A special case of the M-H algorithm is when the proposal does not depend on this preceding step, say $g(y)$. That is, the proposal state y is generated from $g(\cdot)$ independent of the previous step. This kind of proposal was suggested first by Hastings (1970) as an alternative for importance sampling and it is called *Metropolized independence sampler* (MIS).

The MIS algorithm proceeds as follows. It starts in an initial value $x^{(0)}$ such that $\pi(x^{(0)}|\text{data}) > 0$. For $t = 1, 2, \ldots$

1. Generate a proposal y from $g(y)$.
2. Calculate the ratio
$$r = \frac{w(y)}{w(x^{(t-1)})} \tag{2.15}$$
where $w(x) = \pi(x)/g(x)$ is the *importance ratio* or *importance weight*.
3. Set
$$x^{(t)} = \begin{cases} y & \text{with probability } \min(1, r) \\ x^{(t-1)} & \text{otherwise.} \end{cases}$$

Liu (1996) studied the eigenvalues and eigenvectors associated with the transition kernel of the Markov chain for getting bounds of the distance between the target distribution π and the proposal g distribution. This is an exceptional case where the transition matrix can be easily expressed in terms of its eigenvalues, which also can be expanded in terms of cumulative distributions.

The second largest eigenvalue, denoted by λ_1, can be explicitly found in terms of the weights w, and asymptotically controls the mixing rate of the chain, when the state space is finite and the number of iterations is large. An upper bound for the *total variation distance* between g and π is provided in Liu (1996). Additional bounds are given considering other eigenvalues, which also have explicit expressions. See Liu (1996) and Liu (2008).

2.3 Convergence diagnostics

The previous section presented the most common MCMC algorithms. These procedures generate Markov chains which are expected to be representative samples from a target distribution. Assessing the convergence of the Markov chain is one of the challenges when implementing these algorithms. In the MCMC context, convergence refers to answering the question that "at what point is it reasonable to believe that the samples are truly representative of the underlying stationary distribution of the Markov chain?", as Cowles and Carlin (1996, p. 883) discusses. There is an extensive list of convergence diagnostics in the literature which help to identify whether the simulation has been successful in achieving the invariant distribution. Brooks and Roberts (1998) and Cowles and Carlin (1996) provide a thorough compilation of several diagnostics. Four of these convergence diagnostics are briefly presented below. They have been applied in this work via the `CODA` library in the `R` software. See Plummer et al. (2006).

Gelman and Rubin diagnostic

The diagnostic proposed by Gelman and Rubin (1992) applies when multiple Markov chains have been generated. The initial values are assumed to be over-dispersed relative to the posterior distribution. The test identifies convergence when these initial values are forgotten by the chains. This identification is based on a comparison of within-chain and between-chain variances. The diagnostic is briefly explained below.

Suppose M chains with the same length T for a scalar summary θ, with mean μ and variance σ^2 under the target distribution. Let $\bar{\theta}_m$ and s_m^2 be the sample mean and variance of the mth chain, and $\bar{\theta} = (1/M)\sum_{m=1}^{M} \bar{\theta}_m$ the mean over all the chains. There are two ways to estimate σ^2 by using:

1. The mean of the within-chain variances, that is, $W = \dfrac{1}{M}\sum_{m=1}^{M} s_m^2$.
2. The pooled variance which results from combining all the chains. An estimate of this is given by
$$\widehat{V} = \frac{T-1}{T}W + \frac{M+1}{MT}B,$$
where
$$\frac{B}{T} = \frac{1}{M-1}\sum_{m=1}^{M}(\bar{\theta}_m - \bar{\theta})^2$$
represents the between-chain variance.

The *scale reduction factor* is defined by $R = \widehat{V}/\sigma^2$, and it is estimated as $\widehat{R} = \widehat{V}/W$ which is called *potential* scale reduction factor (PSRF). This ratio determines the convergence criterion. If the chains have converged, then both estimates $\widehat{V}$ and W are unbiased. Otherwise, $\widehat{V}$ will overestimate the variance due to overdispersion of the initial states, and W will underestimate the variance since the chains have not covered the support of the stationary distribution. Thus, longer simulations should be executed in order to decrease B or increase W.

Brooks and Gelman (1998) included a correction factor into the original PSRF by accounting for sampling variability in the variance estimates. They define the corrected PSRF as

$$\widehat{R}_c = \frac{d+3}{d+1}\frac{\widehat{V}}{W}$$

where d is the degrees of freedom estimate of a t distribution. Then, convergence is determined if $\widehat{R}_c \approx 1$. Brooks and Gelman (1998) suggested that convergence can be concluded $\widehat{R}_c < 1.2$.

One difficulty with this diagnostic is the assumption of normality of the marginal distribution of each scalar summary θ. This is a strong assumption as "MCMC methods are often used for highly non-normal, and even multimodal densities" as stated by Brooks and Gelman (1998). Also, it requires certain knowledge of the target distribution in order to choose the initial states which are assumed to be over-dispersed.

Geweke diagnostic

The diagnostic proposed by Geweke (1992) compares the means of the sampled parameter at two different parts of the Markov chain, called "windows" (e.g. the first 10% and the last 50%). If the two means are close, then the two corresponding samples come from the same distribution, which is the stationary distribution of the chain. Under this assumption, the Geweke's statistic has an asymptotically standard normal distribution. The test statistic is a standard Z-score which is the difference between the two sample means divided by its estimated standard error. For this estimation, the statistic uses spectral density estimation, which is beyond the scope of this thesis.

To describe briefly Geweke diagnostic, suppose that θ is the parameter of interest. The windows are defined as $\omega_1 = \{k : 1 \leq k \leq t_{\omega_1}\}$ and $\omega_2 = \{k : t^* \leq k \leq t\}$ such that $1 < t_{\omega_1} < t^* < t$ and $\frac{t_{\omega_1}+t_{\omega_2}}{t} < 1$, where $t_{\omega_2} = t - t^* + 1$. Let $\bar{\theta}_{\omega_1}$ and $\bar{\theta}_{\omega_2}$ denote the empirical means at the windows ω_1 and ω_2, which are defined as

$$\bar{\theta}_{\omega_1} = \frac{1}{t_{\omega_1}} \sum_{k \in \omega_1} \theta^{(k)} \qquad \text{and} \qquad \bar{\theta}_{\omega_2} = \frac{1}{t_{\omega_2}} \sum_{k \in \omega_2} \theta^{(k)}.$$

The premise of the diagnostic is to conclude stationarity under the condition that the two sub-samples have been drawn from the same distribution. This is done by testing the null hypothesis that the two means $\bar{\theta}_{\omega_1}$ and $\bar{\theta}_{\omega_2}$ are equal. The statistic is given by

$$Z = \frac{\bar{\theta}_{\omega_1} - \bar{\theta}_{\omega_2}}{\sqrt{\frac{1}{t_{\omega_1}}\hat{S}_{\theta}^{\omega_1}(0) + \frac{1}{t_{\omega_2}}\hat{S}_{\theta}^{\omega_2}(0)}} \to N(0,1) \quad \text{when} \quad t \to \infty$$

where $\hat{S}_{\theta}^{\omega_1}(0)$ and $\hat{S}_{\theta}^{\omega_2}(0)$ are the asymptotic variances of θ_{ω_1} and θ_{ω_2}, respectively, under the assumption of the existence of a spectral density. Thus, the outcome of the command in the `CODA` library is a Z-score to test the hypothesis of means equality.

Heidelberger and Welch diagnostic

This convergence test is based on the procedure suggested by Heidelberger and Welch (1983) for detecting the initial portion of a simulated sequence that contains a transient phase. It tests the null hypothesis that the sampled values come from a stationary distribution by using a Cramer-von-Mises statistic. The test is first applied to the entire chain, and then successively to the sub samples. For example, discarding the first 10%, 20%, 30%, etc. of the chain until either the null hypothesis is accepted, or 50% of the chain has been discarded. The test has two fundamental problems. First, the existence of an initial transient phase which impedes the chain to approximate the steady state characteristics. Second, there are inevitable correlations in the chain.

In `CODA`, the test is divided into two parts: a stationarity test and a half-width test. A failure of the former means that more simulations are needed. If the stationarity test is passed, the number of iterations to be discarded (burn-in) is provided. The half-width test calculates a 95% confidence interval for the mean, using the portion of the chain which passed the stationarity test. First, a target value for the ratio of half-width to sample mean is fixed. Then, half the width of this interval is compared with the estimate of the mean. If the ratio between the half-width and the mean is lower than the target value, the half-width test is passed. Otherwise, the sample is not long enough to accurately estimate the mean.

Raftery and Lewis diagnostic

This diagnostic test uses the quantiles to depict the simulated distribution. For a parameter θ to be monitored, the value u such that $\Pr(\theta \leq u) = q$, for some value $q \in (0,1)$, should be estimated. Raftery and Lewis (1992) proposed a run length control diagnostic based on a criterion of accurately estimating u, that is, the quantile q. The number of iterations required to estimate the quantile q to within an accuracy of r with probability p is calculated. Specifically, $\hat{u} \in [u - r, u + r]$ for some probability p. The diagnostic is briefly described as follows.

A new process $\{Z_t\}$ is defined as $Z_t = \mathbb{1}_{\{\theta^{(t)} \leq u\}}$ where $\mathbb{1}$ is an indicator function. This new process is derived from a Markov chain by "marginalization and truncation, but it is not itself a Markov chain" as explained in Raftery and Lewis (1992). From Z_t, the process $Z_t^{(k)} = Z_{1+k(t-1)}$ is obtained. This can be considered as a lag of Z_t which may behave as a Markov chain. Thus, $Z_t^{(k)}$ is approximately a Markov chain for large

values of k. The required sample size to estimate u is calculated from this thinned sequence.

Autocorrelations and effective sample size (ESS)

Once a sample has been drawn from a target distribution by generating a Markov chain, an important question is how informative the sample is about the parameter. If the correlation between successive or neighbour draws is high, then the chain will not be as informative as in the case of independent simulation draws. The *effective sample size* (ESS) estimates the number of independent simulation draws in the Markov chain. For example, if the simulations in a Markov chain are highly correlated, then 1000 of them may have the same quality as 100 independent simulations. If they are weakly correlated, then 1000 simulations from the chain could be as valuable as 300 independent. The ESS is equal to the length of the number of iterations only if the chain can produce completely uncorrelated samples, which is "possible but in practice highly unlikely", Gelman et al. (2004, p. 299).

Liu (2008) defined an estimator for the effective sample size as

$$\text{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho(k)}$$

where T is the length of the Markov chain and $\rho(k)$ is the correlation at lag k. Other approaches for estimating ESS can be found in see Robert and Casella (2004) and Gelman et al. (2004).

In a time series, the lag-k autocorrelation gives the correlations between pairs of samples that are distanced k iterations away. In other words, it will yield the correlation between a series and its delayed version (time = k). This clearly justifies the use of the word "autocorrelation" since it describes how similar the time series is with itself. It is expected that this autocorrelation becomes smaller as k increases.

The autocorrelations between values generated by a Markov chain can provide clues about its convergence. Although, the Markov chain generated values are dependent, they may be weakly correlated. Large values of autocorrelations for higher values of k indicates a high degree of correlation, which is a signal of slow mixing. The faster the chain mixes, the faster the dependence from the initial state decays in successive iterations, and thus, the faster it converges. Therefore, autocorrelations reveal the mixing speed of the Markov chain.

The `fitR` package (from https://sbfnk.github.io/fitR/index.html), which has been developed for fitting dynamic infectious disease models to time series, includes the `plotESSBurn` function. The output is a plot of the ESS against the burn-in time. It is based on the idea that the ESS is reduced by discarding too many samples (when, in reality, they are informative) or by discarding too few of them (when, in reality, they are not informative). Then, it sounds reasonable the idea of using the ESS estimator, defined above, to have an estimate of the burn-in period length.

Part II

Modelling Uncertainty

Chapter 3

Two MCMC Strategies

As introduced in Section 1.4, Wright et al. (2009) proposed a Bayesian model for estimating population size under the uncertainty of the assignment of identities to the individuals. Wright et al. constructed a Markov chain using the GENUAD algorithm (GENotype Uncertainty by Allelic Dropout) for sampling from the posterior distribution associated with the model.

On the other hand, record or data linkage was introduced in Section 1.5 as an approach to deal with misidentification problems. Specifically, Steorts et al. (2016) provided a Bayesian model for finding the sets of records, across different files, which match a common individual. For sampling from the corresponding posterior distribution, they generated a Markov chain using the SMERED algorithm (Split and MErge REcord linkage and De-duplication).

This chapter presents the models for each of these two approaches. The GENUAD and SMERED algorithms, which were implemented for generating a sample from the corresponding posterior distributions, are explained in detail. Also, illustrative examples have been included.

3.1 Notation

Table 3.1 shows the parameters and the respective distributions to be discussed in this chapter. The adopted notation is that in Wright et al. (2009). The format of the table displays analogous terminology in the same line, for example, the so-called *observed genotypes* g^{obs} in Wright et al. (2009) are named *records* in Steorts et al. (2016). This table unifies the notation of the two approaches.

From Table 3.1, "population size" refers to the number of unique latent individuals in the population and "sample size" to the number of these that were observed in the sample. The definition of these two important concepts may seem redundant at this point. However, the clarity is needed because, as explained in Section 1.4, the presence of allelic dropout adds a genotyping error. Although S DNA profiles are obtained, they may correspond to $n \leq S$ distinct individuals comprising the sample.

Table 3.1: Unifying the notation of GENUAD and SMERED by analogy.

Wright et al. model (GENUAD)	Steorts et al. model (SMERED)
g^{obs}: observed genotypes	Records
$\mathcal{G}$: true genot. in the population (Cat*)	True information in the sample (Cat*)
X: indicator matrix (Cat*)	Indices of the individuals (Uniform)
γ: allele frequencies (Dirichlet)	Multinomial probabilities (Dirichlet)
p: dropout probabilities (Beta)	Distortion probabilities (Beta)
———	Indicator of distortion (Bernoulli)
n: sample size	Sample size
N: population size	———

*Refers to a categorical distribution.

The matrices $\mathcal{G}$ and X define deterministically the true identities of the individuals present in the sample, which are arranged in a matrix denoted by g with the same dimensions as g^{obs}.

Definition 3.1.1. The *true observed genotypes* in g^{obs} are represented by a $S \times L$ matrix denoted by g. For $i \in \{1, \ldots, S\}$, the ith row of g is defined as $g_i = \mathcal{G}_k$ for some value $k \in \{1, \ldots, N\}$ such that $X_{ki} = 1$. The unique genotypes in g are denoted by G. The number of rows of G is n, the number of unique individuals observed in the sample.

Barker et al. (2014) represented the information in the matrix X by using a vector of indices $y = (y_1, \ldots, y_S)'$ such that $y_i \in \{1, \ldots, N\}$ for all $i = 1, \ldots, S$. Specifically, $y_i = k$ if and only if $X_{ki} = 1$. Thus, G and y together are an alternative representation to the pair $\mathcal{G}$ and X for those individuals from the population that appear in the observed sample.

3.2 GENUAD algorithm: A Gibbs sampler

Section 1.4 introduced the model proposed by Wright et al. (2009), which is based on a mark-recapture study. The capture histories of the individuals, the sessions of sampling and time/space considerations are features of this particular kind of model. Based on the description in Barker et al. (2014), the main characteristics of the model in GENUAD are as follows.

1. The complete period of sampling may be divided into sessions which are discrete. Although Wright et al. (2009) considered that the badger droppings were collected in a single session, the sampling occurred throughout ten days. Barker et al. (2014) provides more detail about the consequences of such assumption.

2. At each session, the time and space in which the sampling occurs are considered continuous. Each sample was obtained from one of a set of three latrines. Each latrine corresponds to a badger social group, called a sett. Sampling is assumed to occur in continuous time, even though there are no records of timing. All that

is known is that the sampling occurred at first light to minimise damage to the samples due to the ultraviolet rays.

3. The number of items collected is a random outcome from the experiment. It could depend on the faecal sample deposit rate of the badgers, what they are eating, etc. Thus, the number of samples collected could not be established in advance.

4. There are duplicates in the experiment of which the researcher is not aware. When collecting the faecal pellets, there is no way to determine the unique presence of individuals. Even after processing the DNA samples, it is possible that two closely related individuals have the same DNA profile. For simplicity, this assumption will be called the "twins" effect.

The last assumption allows the presence of twins in the population of size N. However, assuming twins in the sample of size n contradicts Definition 3.1.1. This definition denoted by g the corresponding true genotypes in g^{obs}, and its *unique* genotypes by a matrix G with n rows. That is, two identical rows in G refer to the same individual observed in the sample.

Also, Wright et al. (2009) assumed that the specific population of badgers is closed. That is, there is no immigration/emigration or births/deaths in that population. Since the period of the sampling is short (ten days), in Woodchester Park, the population of badgers is likely to stay the same over ten days than over a more extended period, say six months, when births can happen. According to the Scottish Natural Heritage website:
http://www.snh.org.uk/publications/on-line/naturallyscottish/badgers/breeding.asp,
"all the young are born between mid-January and mid-March, after which they can emerge from the sett to the warmth of the spring". Therefore, that period of ten days can be considered as a short period which allows the assumption of a closed badgers population.

The following small-scale example illustrates the misidentification problem in the badger data. As explained on page 9, g^{obs} comprises observed genotypes, which are represented as pairs of natural numbers, one for each allele (see Definition A.1.1). The notion of compatible genotype given by Definition A.1.3 is also required.

Example 3.2.1. Suppose that a sample of $S = 3$ observed genotypes at $L = 2$ loci with a single replicate is given by

$$g^{\text{obs}} = \begin{pmatrix} 1,1 & 1,1 \\ 1,2 & 2,2 \\ 1,2 & 1,3 \end{pmatrix}.$$

Under the assumption of allelic dropout, there is no certainty that these genotypes represent three distinct individuals. The only reliable information is that sample 3 is known (because of the heterozygotes at the two loci). According to Definition A.1.4, there are three ways of clustering these samples, as shown by Figure 3.1. The edges in

the figure indicate that the samples may belong to the same individual. An example of the first case in Figure 3.1 is when samples 1 and 2 belong to the individual with the true genotype $(1,2 \quad 1,2)$. For the second case, samples 1 and 3 may be associated with the true genotype $(1,2 \quad 1,3)$, while for the third case, all the samples are different. For example, $(1,2 \quad 1,2), (1,2 \quad 2,3)$, and $(1,2 \quad 1,3)$, respectively.

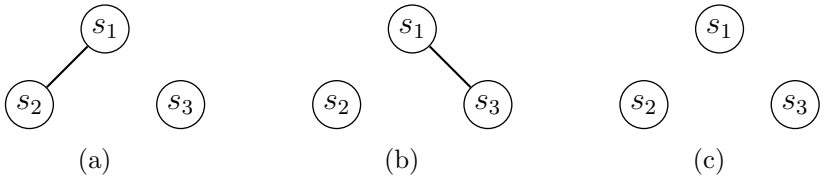


Figure 3.1: Possible clusters for the three samples s_1, s_2 and s_3 in g^{obs}.

The number of alleles detected at each locus is denoted by the vector $m = (2, 3)$. The number of possible genotypes at locus i is given by $\eta_i = m_i(m_i + 1)/2$. Then, $\eta = (3, 6)$. For locus 1, the three possible genotypes are $\{(1,1), (1,2), (2,2)\}$ with probabilities represented by the vector $\gamma^{(1)}$. This notation was taken from Wright et al. (2009), where the superscript indicates the locus. For locus 2, the 6 possible genotypes are $\{(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\}$ with $\gamma^{(2)}$ containing their respective probabilities. These vectors determine the population frequencies of alleles as $\gamma = (\gamma^{(1)}, \gamma^{(2)})'$. Recall that γ, N and p are unknown quantities to be estimated. Table 3.2 shows an example of values for $\gamma^{(1)}$ and $\gamma^{(2)}$ for the possible genotypes. Following Wright et al. (2009), Hardy-Weinberg equilibrium is not assumed, which implies that the only requirement is that the population frequencies of alleles sum 1.0 at each locus.

Table 3.2: An example for $\gamma^{(1)}$ and $\gamma^{(2)}$.

Locus 1		Locus 2	
Genotype	$\gamma^{(1)}$	Genotype	$\gamma^{(2)}$
1,1	1/6	1,1	1/21
1,2	2/6	1,2	2/21
2,2	3/6	1,3	3/21
		2,2	4/21
		2,3	5/21
		3,3	6/21

Fixing values for the other unknown quantities with illustrative purposes, they may be $N = 3$, $p = (0.25, 0.35)$, and

$$\mathcal{G} = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 1,1 & 3,3 \end{pmatrix}, \qquad X = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

These values for $\mathcal{G}$ and X indicate that samples 1 and 3 (first and third columns of X) belong to the same individual, which is identified by the first genotype in $\mathcal{G}$.

While the second genotype in $\mathcal{G}$ was observed in the second sample in g^{obs}, the third genotype was not observed. Then, the three observed genotypes belong to two different individuals (i.e. $n = 2$). From Definition 3.1.1, the true genotypes in g^{obs} are given by

$$g = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 1,2 & 1,3 \end{pmatrix}.$$

□

3.2.1 The algorithm

The Gibbs sampler designed by Wright et al. (2009) generates a Markov chain for which the stationary distribution is the posterior distribution in Eq. (1.1). The inferences about $\mathcal{G}, X, N, \gamma$ and p are based on the S observed genotypes in g^{obs}. Updating γ, p, and N is not difficult as Section 1.4 briefly described. However, updating $\mathcal{G}$ and X is more complicated as the full conditional densities are categorical distributions. Their joint density is given by Eq. (1.2), for fixed values of N, γ and p. Algorithm 1 outlines the Gibbs steps for simulating from this posterior distribution. The following section details the full conditional densities of $\mathcal{G}$ and X.

Algorithm 1 GENUAD (GENotype Uncertainty by Allelic Dropout)

1: **Data:** g^{obs}, N, p and γ
2: **Initializers:** $\mathcal{G}$ and X
3: Update X by using its full conditional density given $\mathcal{G}$, shifting X to X^{new}
4: Update $\mathcal{G}$ by using its full conditional density given X^{new}, shifting $\mathcal{G}$ to $\mathcal{G}^{\text{new}}$
5: **return** $\mathcal{G}^{\text{new}}, X^{\text{new}}$

3.2.2 The Gibbs moves

This section explains the full conditional densities of $\mathcal{G}$ and X based on the supplementary material of Wright et al. (2009). The description of both of them contains imprecisions. While the characterisation of the full conditional of $\mathcal{G}$ seems to have a typographical error, the inaccuracy with the full conditional of X is a fundamental problem since Wright et al. defined an inappropriate model for X. This section will show that the choice of modelling the association between the observations and the actual genotypes by using either an indicator matrix X or a vector of indices y alters the posterior distribution of interest. Section 3.2.2.2 discusses this matter.

3.2.2.1 Full conditional density $f(\mathcal{G}|X, N, \gamma, p)$

The description in this section follows the supplementary material of Wright et al. (2009), where the matrix $\mathcal{G}$ is updated row by row (i.e. by individuals), and locus by locus, given values of X, N, γ and p.

First, consider individuals that appeared at least once in one of the S samples. For individual $i \in \{1, \ldots, N\}$, find c_i, the number of times that i appeared. For locus l, with $l = 1, \ldots, L$, find all genotypes that are compatible (see Definition A.1.3) with the c_i observed genotypes in that locus. Suppose there are $\nu \geq 1$ compatible genotypes. According to Wright et al. (2009) at the supplementary material, the probability of choosing the kth, denoted by $\mathcal{G}_{il}^{(k)}$ for $k = 1, \ldots, \nu$, is defined by $\lambda_{il}^{(k)} / \sum_{h=1}^{\nu} \lambda_{il}^{(h)}$, where

$$\lambda_{il}^{(k)} = \prod_{j=1}^{S} \left(X_{ij} \prod_{r=1}^{R} \Pr(g_{jlr}^{\text{obs}} | \mathcal{G}_{il}^{(k)}, p) \right) \cdot \Pr(\mathcal{G}_{il}^{(k)} | \gamma)$$

Wright et al. (2009) defined the probabilities in this equation, however, the first one can be found in Appendix A.2.1, and the second corresponds to the genotype probabilities (which depend on the population allele frequencies γ). For individuals unseen in the S samples the authors suggest sampling from a categorical distribution.

The problem with this definition of $\lambda_{il}^{(k)}$ is that it is non-zero only if the individual i appears in all samples which is unlikely. This was a typographical error in Wright et al. (2009). Nevertheless, the correct expression was used in the code. Although the following formula fixes the problem, the explanation is still vague.

$$\lambda_{il}^{(k)} = \prod_{j=1}^{S} \left(X_{ij} \prod_{r=1}^{R} \Pr(g_{jlr}^{\text{obs}} | \mathcal{G}_{il}^{(k)}, p) + (1 - X_{ij}) \right) \cdot \Pr(\mathcal{G}_{il}^{(k)} | \gamma) \tag{3.1}$$

If individual i appeared in samples indexed by j in a set of indices $\mathcal{I}$, then Eq. (3.1) takes the form

$$\lambda_{il}^{(k)} = \prod_{j \in \mathcal{I}} \prod_{r=1}^{R} \Pr(g_{jlr}^{\text{obs}} | \mathcal{G}_{il}^{(k)}, p) \cdot \Pr(\mathcal{G}_{il}^{(k)} | \gamma),$$

which is non-zero for all $k = 1, \ldots, \nu$ because all genotypes $\mathcal{G}_{il}^{(k)}$ are compatible with the samples indexed by $j \in \mathcal{I}$. This fact creates a conflict with a statement in the supplementary material: "most of the $\lambda_{il}^{(k)}$ are zero, except for the c_i cases where $X_{ij} = 1$". Besides Eq. (3.1) cannot be used if individual i was not seen in the S samples because defining $\lambda_{il}^{(k)} = \Pr(\mathcal{G}_{il}^{(k)} | \gamma)$ would limit the sampling only to compatible genotypes. Thus, Eq. (3.1) does not solve the problem. The following description endeavours clarify this subject by introducing a different explanation to that in the supplementary material.

For locus l, with $l = 1, \ldots, L$, consider all possible genotypes. Let m_l be the number of alleles at locus l. The number of possible genotypes at that locus is $\eta_l = m_l(m_l+1)/2$. For updating the ith row of $\mathcal{G}$ at locus l, one of the η_l possible genotypes is randomly chosen. The probability for the kth, denoted by $\mathcal{G}_{il}^{(k)}$ for $k = 1, \ldots, \eta_l$, is defined by $\lambda_{il}^{(k)} / \sum_{h=1}^{\eta_l} \lambda_{il}^{(h)}$, where

$$\lambda_{il}^{(k)} = \prod_{j=1}^{S} \left(X_{ij} \prod_{r=1}^{R} \Pr(g_{jlr}^{\text{obs}} | \mathcal{G}_{il}^{(k)}, p) + (1 - X_{ij}) \right) \cdot \Pr(\mathcal{G}_{il}^{(k)} | \gamma) \tag{3.2}$$

If individual i was seen in the samples indexed by $j \in \mathcal{I}$, then Eq. (3.2) is equivalent to

$$\lambda_{il}^{(k)} = \prod_{j \in \mathcal{I}} \prod_{r=1}^{R} \Pr(g_{jlr}^{\text{obs}} | \mathcal{G}_{il}^{(k)}, p) \cdot \Pr(\mathcal{G}_{il}^{(k)} | \gamma). \tag{3.3}$$

Otherwise,

$$\lambda_{il}^{(k)} = \Pr(\mathcal{G}_{il}^{(k)} | \gamma). \tag{3.4}$$

In this case, if individual i does not appear in any of the S samples, or if it appears and all observed genotypes in which does so are compatible with $\mathcal{G}_{il}^{(k)}$, then $\lambda_{il}^{(k)} \neq 0$. If some observed genotype in which the individual i appeared is not compatible with $\mathcal{G}_{il}^{(k)}$, then $\lambda_{il}^{(k)} = 0$. This explanation is more consistent than the description in the supplementary material.

Note that Eqs. (3.1) and (3.2) are the same equations but the reality is that they are different due to the distinct set of genotypes $\{\mathcal{G}_{il}^{(k)}\}$. The former considers the compatible genotypes, while the latter considers all possible genotypes in the locus. The following example illustrates the improved explanation above.

Example 3.2.2. Consider g^{obs}, $\mathcal{G}$ and X as in Example 3.2.1.

$$g^{\text{obs}} = \begin{pmatrix} 1,1 & 1,1 \\ 1,2 & 2,2 \\ \boxed{1,2} & 1,3 \end{pmatrix}, \qquad \mathcal{G} = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 1,1 & 3,3 \end{pmatrix}, \qquad X = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The aim is to update $\mathcal{G}$, given X. The description that involves Eq. (3.2) and considers all possible genotypes in the locus is used.

The number of alleles at each locus is given by $m = (2, 3)$. Suppose that at locus 1, the possible genotypes are in the set $U = \{(1,1), (1,2), (2,2)\}$, and for locus 2, $V = \{(1,1), (1,2), (1,3), (2,2), (2,3), (3,3)\}$. The superscript k in $\lambda_{il}^{(k)}$ refers to the order in these sets. As $\mathcal{G}$ is updated row by row (individual by individual), all the samples in which each individual appeared are identified; and a row is updated locus by locus. For example, the first individual in $\mathcal{G}$ appears in samples 1 and 3. For updating the first row of $\mathcal{G}$ at locus 1, $\lambda_{11}^{(1)} = \lambda_{11}^{(3)} = 0$ because the third observed genotype at locus 1 (inside the square of g^{obs}) is only compatible with the second genotype in U, for which,

$$\lambda_{11}^{(2)} = \prod_{j \in \{1,3\}} \Pr(g_{j1}^{\text{obs}} | \mathcal{G}_{11}^{(2)} = 1,2) \cdot \Pr(\mathcal{G}_{11}^{(2)} = 1,2 | \gamma) \neq 0$$

At locus 2, using the same argument of compatibility, only $\lambda_{12}^{(3)}$ is distinct to zero, and it corresponds to $\mathcal{G}_{12}^{(2)} = 1,3$. Then, $(1,2 \quad 1,3)$ is sampled with probability 1.0 to

update the first row in $\mathcal{G}$.

Now, the second individual in $\mathcal{G}$ appeared in sample 2. At locus 1, only $\lambda_{21}^{(2)} \neq 0$. At locus 2, $\lambda_{22}^{(k)} \neq 0$ for $k = 2, 4, 5$, otherwise zero. This is because the other genotypes in V are not compatible with 2,2 in the second observed genotype in g^{obs} at the second locus.

$$\begin{aligned}
\lambda_{22}^{(2)} &= \Pr(g_{22}^{\text{obs}} = 2,2|\mathcal{G}_{22}^{(1)} = 1,2) \cdot \Pr(\mathcal{G}_{22}^{(1)} = 1,2|\gamma^{(2)}) &&= (p_2/2) \cdot (2/21) \\
\lambda_{22}^{(4)} &= \Pr(g_{22}^{\text{obs}} = 2,2|\mathcal{G}_{22}^{(2)} = 2,2) \cdot \Pr(\mathcal{G}_{22}^{(2)} = 2,2|\gamma^{(2)}) &&= 1 \cdot (4/21) \\
\lambda_{22}^{(5)} &= \Pr(g_{22}^{\text{obs}} = 2,2|\mathcal{G}_{22}^{(3)} = 2,3) \cdot \Pr(\mathcal{G}_{22}^{(3)} = 2,3|\gamma^{(2)}) &&= (p_2/2) \cdot (5/21)
\end{aligned}$$

where $p = (p_1, p_2)$ contains the dropout probabilities at each loci, and $\gamma = (\gamma^{(1)}, \gamma^{(2)})$ as in Table 3.2. Suppose that $(1,2)$ is drawn with probability $\frac{\lambda_{22}^{(2)}}{\cdot \sum_{h=1}^{6} \lambda_{22}^{(h)}}$. Thus, the new second row of $\mathcal{G}$ is $(1,2 \quad 1,2)$ with probability $2p_2/(7p_2 + 8)$.

As the third individual was not seen in the sample, it is sampled from a categorical distribution, whose multinomial probabilities are given by $\Pr(\mathcal{G}|\gamma)$. Suppose that $(1,1 \quad 1,2)$ has been sampled with probability equal to $\left(\frac{1}{6}\right)\left(\frac{2}{21}\right)$. Then, updating $\mathcal{G}$ results in a new matrix $\mathcal{G}^*$ as follows.

$$\mathcal{G}^* = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 1,2 \\ 1,1 & 1,2 \end{pmatrix} \tag{3.5}$$

□

3.2.2.2 Full conditional density $f(X|\mathcal{G}, N, \gamma, p)$

This section aims to describe an appropriate procedure for updating X. It makes adjustments to the description in Wright et al. (2009) and its supplementary material.

Given $\mathcal{G}, N, \gamma$, and p, the indicator matrix X is updated column by column (observation by observation). For updating the jth column of X, for $j = 1, \ldots, S$, the strategy consists of substituting the jth row of g (the matrix of true genotypes in the sample) with each row of $\mathcal{G}$. If row j of g is replaced by row i of $\mathcal{G}$, the resulting matrix is denoted by $g^{(i)}$. In general, these exchanges result in a set of possible matrices $\Psi_j = \{g^{(i)} : i = 1, \ldots, N\}$ differing only in the jth row. Each of these matrices defines an indicator matrix $X^{(i)}$ associated with the ith switch. As in section 1.3 of the supplementary material of Wright et al. (2009),

$$\lambda_{ji} = \Pr(g_j^{\text{obs}}|g_j^{(i)}, p) \cdot \Pr(X^{(i)}|N) \tag{3.6}$$

where the first factor is computed using the probability in Appendix A.2.1 and the second is modelled by

$$f(X|N) = \frac{N!}{n!\,(N-n)!} \frac{S!}{c_1! \ldots, c_n!} \left(\frac{1}{N}\right)^S$$

where c_i is the number of times that individual i appeared in the S samples. This expression is the equation (6) in Wright et al. (2009, p. 836). However, $f(X|N)$ should be $1/N$. This assertion is based on the simplest case in which all individuals in the population are equally likely to be associated with the jth observed genotype.

From Ψ_j, one matrix is chosen with probability $\lambda_{ji}/\sum_{h=1}^{N}\lambda_{jh}$. The corresponding indicator matrix is the new value of X. Note that this ratio implies that the term $1/N$ cancels out. Then, that term is not required for updating X. Also, $\lambda_{ji}=0$ if the ith row of $\mathcal{G}$ is not compatible with g_j^{obs}.

Updating X may result in a different value of n (the number of unique individuals observed in the sample) because the number of zero rows associated to those that were not observed, could be reduced or increased when the location of the 1's switches in a column of X.

Example 3.2.3. Continuing with Example 3.2.2, given $\mathcal{G}^*$ in Eq. (3.5), the matrix X is updated column by column. For the first observed genotype, the matrices in Ψ_1 are,

$$g^{(1)}=\begin{pmatrix}\boxed{1,2 & 1,3}\\ 1,2 & 1,2\\ 1,2 & 1,3\end{pmatrix},\quad g^{(2)}=\begin{pmatrix}\boxed{1,2 & 1,2}\\ 1,2 & 1,2\\ 1,2 & 1,3\end{pmatrix},\quad g^{(3)}=\begin{pmatrix}\boxed{1,1 & 1,2}\\ 1,2 & 1,2\\ 1,2 & 1,3\end{pmatrix},$$

in which the possible values for the first row are indicated. The corresponding X matrices are,

$$X^{(1)}=\begin{pmatrix}1&0&1\\0&1&0\\0&0&0\end{pmatrix},\quad X^{(2)}=\begin{pmatrix}0&0&1\\1&1&0\\0&0&0\end{pmatrix},\quad X^{(3)}=\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}.$$

According to the explanation above, the probabilities are calculated as follows.

$$\begin{aligned}\lambda_{11}=\Pr(g_1^{\text{obs}}=(1,1\quad 1,1)|g_1^{(1)}=(1,2\quad 1,3))/N &= \frac{p_1}{2}\cdot\frac{p_2}{2}\cdot\frac{1}{3}\\ \lambda_{12}=\Pr(g_1^{\text{obs}}=(1,1\quad 1,1)|g_1^{(2)}=(1,2\quad 1,2))/N &= \frac{p_1}{2}\cdot\frac{p_2}{2}\cdot\frac{1}{3}\\ \lambda_{13}=\Pr(g_1^{\text{obs}}=(1,1\quad 1,1)|g_1^{(3)}=(1,1\quad 1,2))/N &= 1\cdot\frac{p_2}{2}\cdot\frac{1}{3}\end{aligned}$$

Thus, one element from Ψ_1 is sampled with probability equal to $\lambda_{1i}/\sum_{h=1}^{3}\lambda_{1h}$ for $i=1,2,3$, and it determines the choice of the indicator matrix. This process applies to all the samples in g^{obs} until all the columns of X have been updated. Sample 2 is only compatible with $\mathcal{G}_2^*=(1,2\quad 1,2)$, thus Ψ_2 only has a single element which is chosen with probability 1.0. Sample 3 is similar, which is only compatible with $\mathcal{G}_1^*=(1,2\quad 1,3)$. Suppose, then, that the new value of X, given $\mathcal{G}^*$, is

$$X^*=\begin{pmatrix}0&0&1\\0&1&0\\1&0&0\end{pmatrix}$$

Notice that the value of n has been updated, the new value is $n^*=3$.

□

3.2.3 Modelling the vector of indices y

The full conditional density of X is a critical element of GENUAD. It not only indicates which individual an observed genotype belongs to, but also it determines how many unique individuals were observed in the sample. Section 3.2.2.2 described an appropriate procedure for updating X. This section discusses why the choice of the model for X in Wright et al. (2009) was inaccurate.

Wright et al. followed Miller et al. (2005) in assuming that the genotypes, used as tags for uniquely identifying the individuals, are drawn from the population one at a time and with replacement. The estimation of abundance, in Miller et al. (2005), is based on the log-likelihood function given by

$$L(N) = \frac{N!}{n!\,(N-n)!} \cdot \frac{S!}{c_1! \ldots c_n!} \prod_{i=1}^{n} \left(\frac{1}{N}\right)^{c_i} \tag{3.7}$$

where N is the population size, S is the sample size, n is the number of distinct individuals in the sample, and c_i is the number of times that individual i is seen in the sample, with $S = \sum_{i=1}^{n} c_i$. It has been assumed that individuals have the same probability of appearing in the sample which is equal to $1/N$. However, Bromaghin (2007) argued that the likelihood is not correct due to the absence of labels for differentiating similar capture histories of the individuals, who are distinguishable. The Bromaghin argument is that "this function is not a valid likelihood function as the sum of all possible outcomes does not sum to 1.0".

Bromaghin (2007) introduced an additional term to Eq. (3.7) resulting in the correct likelihood given by

$$L(N) = \frac{N!}{n!\,(N-n)!} \cdot \frac{S!}{c_1! \ldots c_n!} \cdot \frac{n!}{u_1! \ldots u_U!} \left(\frac{1}{N}\right)^{S} \tag{3.8}$$

where u_i is the number of times c_i appears in $c = (c_1, \ldots, c_n)$, U the number of unique values in c, and $n = \sum_{i=1}^{U} u_i$. This term takes into account the multiplicities of the counts c_i, $i = 1, \ldots, n$, and their different arrays. Bromaghin used an example to show that the sum of all outcomes is equal to N^S when using this formula, but not when using Miller's formula.

However, neither Miller et al. (2005) or Bromaghin (2007) fit the model of X in Wright et al. (2009). The reason lies in the choice between X and y for establishing the connections between the observations and the latent individuals. If X, then the model is as Section 3.2.2.2 described; if y, the model is different, as explained below.

Barker et al. (2014) considered the data in Wright et al. for describing a general capture-recapture model because they were interested in samples which are drawn one at time. They denoted the identities of the individuals appearing in the sample j, for

$j = 1, \ldots, S$, as $y_j \in \{1, \ldots, N\}$. These elements are the components of the vector of indices y, and according to the equation 4 in Barker et al. (2014), it is modelled by

$$f(y|N) = \frac{N!}{(N-n)!}\left(\frac{1}{N}\right)^S \tag{3.9}$$

where n is the number of unique indices in y. An important remark is that once an item is drawn, the index i is unknown. The use of labels (for example A, B,...) is then convenient, as explained by Barker et al. (2014). Following their notation, an observed history is a sequence of labels of length S. For example, the vectors $y = (1, 2, 1, 1)'$ and $y^* = (2, 1, 2, 2)'$ are represented by the same observed history ABAA. As follows, the density in Eq. (3.9) provides the probability of an observed history.

To provide an intuitive derivation of the density in Eq. (3.9), an observed history with size S is denoted by $\mathcal{H}$. The question is how many vectors y with n unique elements taken from a set of N, with $n \leq N$, can be represented with $\mathcal{H}$. To answer that question, consider y as an empty list. The first component has N possible values, the second unique has $N-1$ possibilities, the third unique has $N-2$, and continuing with this process, there are $N-(n-1)$ for the nth unique component. Applying the multiplication principle, the empty list y can be filled in $N(N-1)\cdots(N-n+1) = N!/(N-n)!$ possible ways. That is, there are $N!/(N-n)!$ arrays y (with n unique components) which are represented by the observed history $\mathcal{H}$. Each of them has probability of occurrence equal to $(1/N)^S$ because all N elements are equally likely to appear in the vector y with length S. Then, the product $N!/(N-n)! \cdot (1/N)^S$ gives the probability of $\mathcal{H}$.

A probability problem of rolling a die illustrates the use of the density in Eq. (3.9). There are two players. Player one has a six-sided die, and player two is blindfolded. Player one rolls the die five times, writing down the outcome each time. The only information that player two receives from player one is that the first two outcomes are the same, and the other three are different amongst themselves and different from the first two. What is the probability that player two can guess the result? The sample space of this experiment has sequences of five numbers with the form AABCD, which resemble the capture histories mentioned before. Thus, Eq. (3.9) solves this problem.

Therefore, indicator matrices X in Wright et al. (2009) and the vector of indices y in Barker et al. (2014) are modelled differently even though, when paired with $\mathcal{G}$ they provide the same information about who was observed in which DNA sample.

Example 3.2.4. Consider the indicator matrix X as below.

$$X = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

This indicator matrix reports the presence of $n = 2$ unique individuals in the observed sample from a population of $N = 6$. Also, it indicates that an individual indexed by 1 appears in the first, third, and fourth samples, while the individual indexed by 2 appears in the second sample. In other words, it is equivalent to the vector of indices $y = (1, 2, 1, 1)'$. The matrix X is the result of permutations of the rows of equivalent indicator matrices. Thus, there is no difference between $y = (1, 2, 1, 1)'$ and $y^* = (2, 1, 2, 2)'$. The observed history of interest is ABAA. As follows,

$$\Pr(\text{ABAA}|N = 6) = \frac{6!}{(6-2)!}\left(\frac{1}{6}\right)^4 = 30\left(\frac{1}{6}\right)^4$$

When applying Bromaghin's formula in Eq. (3.8), it gives the probability of observing c_1 and c_2 number of captures, no matter the sample in which the capture occurred. For example,

$$\Pr(c_1 = 3, c_2 = 1|N = 6) = \frac{6!}{2!\,(6-2)!} \cdot \frac{4!}{3!\,1!} \cdot \frac{2!}{1!\,1!}\left(\frac{1}{6}\right)^4 = 120\left(\frac{1}{6}\right)^4$$

which is the probability that individual 1 appears in the sample three times and individual 2 only appears once. Evidently, this is not the desired probability. Note that the event $\{c_1 = 3, c_2 = 1\}$ represents any of the vectors in

$$\{(1, 1, 1, 2), (1, 1, 2, 1), (1, 2, 1, 1), (2, 1, 1, 1)\}$$

The observed histories representing these vectors are AAAB, AABA, ABAA, and BAAA, which are distinct. □

3.3 SMERED algorithm: A Metropolis sampler

Steorts et al. (2016) proposed the SMERED (Split and MErge REcord linkage and De-duplication) algorithm for linking records that could be associated to the same latent individual across different files. It is a Metropolis-within-Gibbs algorithm whose proposals result from split-merge operations, which are explained in the next section. The unification in Table 3.1 avoids the introduction of unnecessary notation, which facilitates reading through this dissertation. SMERED uses the vector of indices y, instead GENUAD uses the indicator matrix X.

3.3.1 Data and model

The dataset in Steorts et al. (2016) is comprised of k *files*. Each file contains n_i records of individuals, which may be distorted, in L common attributes (called *fields*). For file i, $g_{ij\ell}^{\text{obs}}$ represents the value for record j at field ℓ, for $i = 1, \ldots, k$, $j = 1, \ldots, n_i$, and $\ell = 1, \ldots, L$. The records and their true attributes are linked by using a $n \times L$ matrix G and a ragged array y (called *linkage structure*). That is, the latent individual associated to $g_{ij\ell}^{\text{obs}}$ is indexed by y_{ij}, and the corresponding true value is $G_{y_{ij}\ell}$. Whether

that observation has an error or not is indicated by $z_{ij\ell}$. If $z_{ij\ell} = 1$, the value in the ℓth field for the jth record in file i is distorted. Otherwise, there is no distortion.

The model is based on two assumptions of independence:

- Given the latent individuals, the k files are conditionally independent.
- The fields are independent within individuals.

The full Bayesian model proposed by Steorts et al. (2016) is given by:

$$
\begin{aligned}
g_{ij\ell}^{\text{obs}} | G_{y_{ij}\ell}, y_j, z_{ij\ell}, \gamma_\ell &\overset{\text{ind}}{\sim} \begin{cases} \delta_{G_{y_{ij}\ell}} & \text{if} \quad z_{ij\ell} = 0 \\ \text{MN}(1, \gamma_\ell) & \text{if} \quad z_{ij\ell} = 1 \end{cases} \\
G_{y_{ij}\ell} | \gamma_\ell &\overset{\text{ind}}{\sim} \text{MN}(1, \gamma_\ell) \\
\gamma_\ell &\overset{\text{ind}}{\sim} \text{Dirichlet}(\mu_\ell) \\
z_{ij\ell} &\overset{\text{ind}}{\sim} \text{Bernoulli}(p_\ell) \\
p_\ell &\overset{\text{ind}}{\sim} \text{Beta}(a_\ell, b_\ell) \\
\pi(y) &\propto 1
\end{aligned}
$$

The symbol δ_a represents a point mass at a, γ_ℓ denote the multinomial probabilities, and p_ℓ the Bernoulli probabilities. The values a_ℓ, b_ℓ and μ_ℓ are known. The notation for the prior distribution of y $\pi(y) \propto 1$ is slightly unclear. But it seems to refer to a uniform distribution.

The joint posterior distribution $\pi(G, y, z, \gamma, p | g^{\text{obs}})$ is the target distribution when implementing the SMERED algorithm. The full conditional densities of the parameters for applying the Gibbs sampling are given in Steorts et al. (2016, p. 1662).

3.3.2 The algorithm

As stated above, SMERED is a Metropolis-within-Gibbs algorithm. This section presents a description of SMERED according to the additional material in Steorts et al. (2016) which offers more details about the algorithm itself.

Let $G^{(t)}, y^{(t)}, z^{(t)}, \gamma^{(t)}$ and $p^{(t)}$ denote the values for the parameters at step t.

1. Draw a pair of records at random.
2. Propose a split or merge depending on whether they refer to the same individual or not. The proposals are denoted by G' and y'.
3. Calculate r as

$$
r = \frac{\pi(G', y', z^{(t)}, \gamma^{(t)}, p^{(t)}) | g^{\text{obs}})}{\pi(G^{(t)}, y^{(t)}, z^{(t)}, \gamma^{(t)}, p^{(t)} | g^{\text{obs}})}
$$

4. Accept the proposals G' and y' with probability $\min(1, r)$. This means that $y^{(t+1)} = y'$. However, $G^{(t+1)} \neq G'$.

5. Sample $G^{(t+1)} \sim f(G|y^{(t+1)}, z^{(t)}, \gamma^{(t)}, p^{(t)}, g^{\text{obs}})$.

6. Sample $z^{(t+1)} \sim f(z|G^{(t+1)}, y^{(t+1)}, \gamma^{(t)}, p^{(t)}, g^{\text{obs}})$.

7. Sample $\gamma^{(t+1)} \sim f(\gamma|z^{(t+1)}, G^{(t+1)}, y^{(t+1)}, p^{(t)}, g^{\text{obs}})$.

8. Sample $p^{(t+1)} \sim f(p|\gamma^{(t+1)}, z^{(t+1)}, G^{(t+1)}, y^{(t+1)}, g^{\text{obs}})$.

Steps 1-4 simultaneously update y and G by a Metropolis step. This step is executed by applying the split-merge operations. Gibbs sampling is applied for updating G, z, γ and p in steps 5-8. Note that G is updated twice. It is required for updating y and calculating the ratio r.

3.3.3 The split-merge moves

In SMERED, the split-merge operations can be viewed as a joint updater of y and G, which occurs in the Metropolis step. The algorithm starts by randomly choosing a pair of records from different files. If they are associated with the same individual, then a split is proposed. Otherwise, they are merged. Following the supplementary material of Steorts et al. (2016), the steps are outlined as follows.

Splitting

i. Suppose both records are associated to the j_1th latent individual in G.

ii. A new index is created, say j_2.

iii. Identify the set of all records currently assigned to j_1, denoted by C. This set includes the two chosen records.

iv. The two chosen records are randomly assigned to j_1 and j_2 such that one record continues to be assigned to j_1 while the other to j_2.

v. The remaining records in C are randomly assigned to either index j_1 or j_2. This procedure partitions C into two disjoint subsets, C_1 and C_2.

vi. To provide the corresponding values in G, the records in C_1 and C_2 are assumed to be undistorted (free of error). One record from each subset is chosen randomly and established as the new values for the j_1th and j_2th rows in G such that they are different, that is, $G_{j_1} \neq G_{j_2}$.

Thus, the two chosen records which were originally associated with the same latent individual, now match different individuals. Now the merge operation is explained.

Merging

i. Suppose both records are associated with different latent individuals, say j_1 and j_2.

ii. The new index is a random choice from these two indices, say j^*. That is, either $j^* = j_1$ or $j^* = j_2$. The other index is removed.

iii. Denotes the set of all records associated with j_1 and j_2 by C, including the chosen records.

iv. To provide the value G_{j^*}, one record is randomly chosen from C. As before, these records have been assumed without distortion.

Therefore, the two records, originally associated with different individuals, have been merged into a single individual.

The next example, which is a modification of the motivating example used by Steorts et al. (2016), illustrates the process described above. Later in this chapter, the SMERED algorithm is applied to the genotype uncertainty problem considered by Wright et al. (2009).

Example 3.3.1. Consider that the observed data g^{obs} comprises records in three files as shown below. They contain information on address (U.S. state abbreviations), age and gender of individuals. There is no way to identify the individuals due to privacy policies. Unlike the original example, for simplicity, it is assumed here that all the ages are at risk of distortion.

$$\text{File 1} = \begin{pmatrix} \text{NC} & 72 & \text{F} \\ \text{SC} & 70 & \text{F} \\ \text{PA} & 91 & \text{M} \end{pmatrix} \quad \text{File 2} = \begin{pmatrix} \text{SC} & 37 & \text{F} \\ \text{VA} & 93 & \text{M} \\ \text{PA} & 92 & \text{M} \end{pmatrix} \quad \text{File 3} = \begin{pmatrix} \text{NC} & 72 & \text{F} \\ \text{NC} & 72 & \text{F} \\ \text{SC} & 72 & \text{F} \\ \text{VA} & 94 & \text{M} \end{pmatrix}$$

Owing to the presence of distortion, there is no certainty about the actual number of different records and their actual values. Suppose G and y as below. In this case, y is a ragged array that comprises three vectors of indices (one for each file). For example, the first two records in file 3 belong to the same individual, which is indexed by 1 (i.e. the first individual in G). Also, the second and last individuals in G appear in file 3.

$$G = \begin{pmatrix} \text{NC} & 72 & \text{F} \\ \text{SC} & 73 & \text{F} \\ \text{PA} & 91 & \text{M} \\ \text{VA} & 94 & \text{M} \end{pmatrix} \qquad y = \begin{pmatrix} 1 & 2 & 3 & \\ 2 & 4 & 3 & \\ 1 & 1 & 2 & 4 \end{pmatrix}$$

The split-merge operations start with the choice of a pair of records from different files. Suppose for example:

- record 2 from file 1, that is, $(SC, 70, F)$ and
- record 3 from file 3, that is, $(SC, 72, F)$.

The array y shows that these records match with the same latent individual which is the second row in G (using matrix notation, $y_{12} = y_{33} = 2$). Then, a split is required. The steps described above are replicated here to facilitate understanding.

i. Both records are currently associated with the individual 2.

ii. The new index can be set as 5 because currently, 4 is the maximum index. The index can be any integer greater than 4.

iii. The set of all records assigned to latent individual 2 is

$$C = \{(SC, 70, F), (SC, 37, F), (SC, 72, F)\}.$$

iv. The second record from file 1 is assigned to index 5, while the other one stays assigned to index 2.

v. The record 1 in file 2, which is the remaining record, is randomly assigned to one of the indices 2 or 5, say index 2. Then, the sets $C_2 = \{(SC, 37, F), (SC, 72, F)\}$ and $C_5 = \{(SC, 70, F)\}$ have been constructed for indices 2 and 5, respectively.

vi. The values in G for the new indices are chosen randomly from the sets C_2 and C_5, assuming no distortion in those records. For example, $G_2 = (SC, 37, F)$ and $G_5 = (SC, 70, F)$.

This split process has updated the latent information, providing new proposals given by:

$$G' = \begin{pmatrix} \text{NC} & 72 & \text{F} \\ \text{SC} & 37 & \text{F} \\ \text{PA} & 91 & \text{M} \\ \text{VA} & 94 & \text{M} \\ \text{SC} & 70 & \text{F} \end{pmatrix} \qquad y' = \begin{pmatrix} 1 & 5 & 3 & \\ 2 & 4 & 3 & \\ 1 & 1 & 2 & 4 \end{pmatrix}$$

□

Notice that GENUAD and SMERED both update g, the genotypes in the sample. Then, the target distribution given by Eq. (1.2) is rewritten in terms of the observable as,

$$\pi(G, \Lambda | g^{\text{obs}}, N, \gamma, p) \propto f(g^{\text{obs}} | G, \Lambda, p) \cdot f(G | N, \gamma) \cdot f(\Lambda | N) \tag{3.10}$$

where Λ is either X (as in GENUAD) or y (as in SMERED). As seen, their distributions are different. The SMERED algorithm is outlined by Algorithm 2.

3.4 GENUAD vs SMERED: A comparison

A comparison of the approaches in Wright et al. (2009) and Steorts et al. (2016) reveals their similarities and differences. A contrast of the two approaches may show how SMERED can be used for sampling from Eq. (3.10). It can also reveal some differences in how the two algorithms may perform. For simplicity, this section will refer to these two approaches by using the names of their respective MCMC algorithms (i.e. GENUAD and SMERED).

Algorithm 2 SMERED (Split and MErge REcord linkage and De-duplication)

Data: g^{obs}, N, p and γ
Initializers: G and y
Draw a pair of observations, say i and j for some $i \neq j$ in $\{1, \ldots, S\}$ at random.
if $y_i = y_j$ **then**
    Propose splitting that individual, shifting y to y^*
else
    Propose merging the individuals who i and j refer to, shifting y to y^*
end if
Update G using the observations, shifting G to G^*
Calculate $r = \min(1, \pi(G^*, y^* | g^{\text{obs}}, N, p, \gamma) / \pi(G, y | g^{\text{obs}}, N, p, \gamma))$
Set $y^{\text{new}} = y^*$ with probability $\min(1, r)$. Otherwise, set $y^{\text{new}} = y$
Update G^* by using its full conditional density given y^{new}, shifting G^* to G^{new}
return $G^{\text{new}}, y^{\text{new}}$

3.4.1 Similarities

A first affinity between the two approaches is the adoption of a Bayesian model to deal with the uncertainty about the linkage between observations and latent individuals. Bayesian inference for these unknown parameters is conditioned on observed genotypes in the case of GENUAD, and personal information of people in SMERED. Even the parameters considered in both approaches are highly similar as shown by Table 3.1. The actual values for the latent individuals in the sample are drawn using a multinomial distribution with probabilities following a Dirichlet distribution in both cases. Also, the probabilities associated with the source of error in the data have an a priori Beta distribution. The assumptions in the models are also similar. Likewise in SMERED the fields are assumed to be independent. Independence is also assumed between loci in GENUAD.

Both approaches propose a model for directly linking what is observed (genotypes/records) with the true entity that the observation represents and possibly distorts. It implies that the observations are linked indirectly to each other. In SMERED, a set of records associated with the same latent individual are called *co-referent*, and the same term can be used in GENUAD context. That is, a set of genotypes that are compatible with a common latent genotype will be called co-referent (see Definition A.1.4). The concept of compatibility in Definition A.1.3 has been established between observed and latent genotypes.

Also, both procedures allow the estimation of attributes of the observed individuals from the population. For example, the number of unique observations in the sample n, which is given by the number of unique indices in y. Another outcome is the *duplicate* detection which is "the task of identifying and matching records that refer to the same entities within a single database" (Christen, 2012a, p. 4). In general, the *de-duplication* is comprised of merging two or more co-referent records into a single record which accurately represents the entity. While SMERED allows de-duplication

by applying the merge operations, GENUAD updates an indicator matrix which also merge genotypes.

Additionally, the inclusion of categorical variables is allowed in both models. In the case of SMERED, variables as the date of birth, gender, residence state, etc. are considered. While for GENUAD, the variables are the markers at each locus, which are represented by using labels (pairs of positive integers).

3.4.2 Differences

One of the differences between the approaches is the framework they fit. While mark-capture studies are the foundations for establishing the model in GENUAD, record linkage and de-duplication motivated the SMERED model. In general, record linkage allows the verification of the integrity and veracity within and between the data sources, and capture-recapture statistical methods allow for the recovery of estimates of the number of cases missed. In this sense, they could be used together in some analyses. However, GENUAD itself establishes a mechanism for verifying the authentic genotypes in the sample, which adds an advantage to it.

Both SMERED and GENUAD have a probabilistic structure, rather than a deterministic one, as they fit a Bayesian framework. However, Steorts et al. (2016) propose a range of posterior matching probabilities not explored in the case of GENUAD. The posterior probability of linkage between a set of arbitrary records, the posterior probability that a given set of records is linked, and the posterior probability that a given set of records is a maximal matching set. This matching system allows the representation of the pattern of links between co-referent records by using bipartite graphs where the nodes represent the records, and the edges signify the co-reference notion.

Blocking is a concept in record linkage considered in SMERED. It is a technique that partitions the set of records into subsets, called blocks, according to a rule or a systematic process. Steorts et al. (2016) applies approximate blocking with the intention of reducing the number of possible matchings between records. In GENUAD, the concept of compatibility between genotypes limits the number of possible links. Thus, compatibility may be considered as blocking; however, this is a concept that needs further exploration. The "file" notion in SMERED is another concept that does not exist in GENUAD. However, the data in GENUAD could represent a database comprised of S files, each with a single observation. Considering the data as one file with S observations does not make sense because, for the split-merge operations, the pairs of observations are taken from different files.

The corruption or error in the data is different for each approach. It refers to the factor $f(g^{\text{obs}}|G, \Lambda, p)$ in Eq. (3.10). While allelic dropout causes the genotyping error in the data modelled by GENUAD, the source of distortion considered by SMERED is caused by unintended changes to the original data when combining and processing different data sources.

Regarding the algorithms, notice that step 4 in Algorithm 1, and step 12 in Algorithm 2, are almost the same. The difference is that the former updates attributes in the population, while the latter focus on the sample. Thus, the difference between the algorithms is how they update Λ in Eq. (3.10). Indeed, GENUAD updates X using its full conditional density. SMERED updates the indices in y using a Metropolis step which includes the split-merge operations for proposing values of y and G simultaneously. This happens between steps 3 and 11 in Algorithm 2. In this sense, it seems that SMERED could be incorporated in GENUAD for updating the attributes of those individuals that were observed in the sample, as mentioned before.

A crucial dissimilarity is that the SMERED algorithm is inapplicable to the data in GENUAD. Step 9 in Algorithm 2 updates the values in G by using the observations. For the data in GENUAD, this would lead to states which do not belong to the support of the joint density of (G, X). Then, step 9 in Algorithm 2 is an impossible move for the data in GENUAD, and the notion of compatibility in Definition A.1.3 is the reason. For illustrating this, consider the next example which follows up Example 3.2.1.

Example 3.4.1. Consider the observed and the latent genotypes as in Example 3.2.1. Following the steps of the split-merge operations, suppose that the samples 1 and 2 were randomly chosen. According to $\mathcal{G}$ and X, they belong to different individuals in the population, indexed by 1 and 2, respectively, which suggests they can be merged. As illustrated in Example 3.3.1, merging records requires the construction of a set C containing all the samples that belong to latent individuals 1 and 2, from which one is randomly sampled. In this case, $C = g^{\text{obs}}$. Suppose that $(1,1 \quad 1,1) \in C$ was randomly chosen and assigned to index 1. Notice that the assignment of an index does not assume allelic dropout. After merging, the new latent information is given by

$$\mathcal{G}^* = \begin{pmatrix} 1,1 & 1,1 \\ 1,2 & 2,3 \\ 1,1 & 3,3 \end{pmatrix} \qquad X^* = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The problem with the pair $(\mathcal{G}^*, X^*)$ is that it does not belong to the support of the joint density $f(\mathcal{G}, X)$. This is because the three observed genotypes in g^{obs} cannot be co-referent (see Definition A.1.4). □

This example shows that the SMERED model cannot be implemented for the data considered in GENUAD due to the compatibility notion. Then, the SMERED algorithm proposed by Steorts et al. (2016) requires a modification for modelling the data considered in Wright et al. (2009).

Lastly, a significant disparity between the approaches is the notion of population. It is clear that GENUAD updates $\mathcal{G}$, the true genotypes in the entire population, while in SMERED, the role of the population size is not clear. In fact, it seems that the population size and the sample size are assumed to be equal. Indeed, the authors add this caveat, which is discussed below.

3.4.3 The population size in SMERED

In SMERED, the choice of the definition of the population size is a problem that needs further exploration and complicates the comparison with GENUAD. Although section 6.1 in Steorts et al. (2016) started a discussion on this topic, it is reviewed to bring attention to the formula used.

Steorts et al. (2016) defined a *partition* ξ as an array that divides the records in $|\xi|$ latent individuals. If N represents the number of latent individuals and S the number of records, then a prior on ξ is given by

$$\pi(\xi) = \frac{N!}{(N-|\xi|)!}\left(\frac{1}{N}\right)^S, \quad |\xi| \leq S \tag{3.11}$$

where $N!/(N-|\xi|)!$ counts the number of unique y vectors associated to the partition ξ. Eq. (3.11) is a corrected version of the expression given by Steorts et al. (2016, p. 1670)[1]. The authors state that the records are treated as "if they are a random sample drawn *with* replacement". This means that for each record the probability of being randomly chosen is equal to $1/N$, and there are S records. Then, the term $(1/N)^S$ indicates that all the individuals have the same probability of being chosen in the sample.

To illustrate what the partitions are, as before, consider $N = 6$ individuals in a population, and $S = 4$ records in a single file given by $x = (x_1, x_2, x_3, x_4)$. The partition $\xi = \{\{x_1, x_3, x_4\}, \{x_2\}\}$ indicates that records 1, 3 and 4 belong to the same latent individual, which is different to the individual associated with record 2. Eq. (3.11) computes the probability of the partition ξ. Because $|\xi| = 2$ there are $6!/(6-2)! = 30$ ways of choosing two individuals from the population and assign them to the records accordingly to the partition ξ. Besides, all three individuals in the population have the same probability. Then, the probability of the partition ξ is equal to $30/6^4$. By using the observed histories, the partition ξ is equivalent to the history ABAA in Example 3.2.4.

As discussed above, the Bromaghin's formula in Eq. (3.8) gives the probability of $c_1 = 1$ and $c_2 = 3$ which includes the partitions in Table 3.3. The corresponding observed histories appear in the second column.

Table 3.3: Partitions associated to the event $\{c_1 = 1, c_2 = 3\}$.

ξ	Observed history
$\{\{x_2, x_3, x_4\}, \{x_1\}\}$	BAAA
$\{\{x_1, x_3, x_4\}, \{x_2\}\}$	ABAA
$\{\{x_1, x_2, x_4\}, \{x_3\}\}$	AABA
$\{\{x_1, x_2, x_3\}, \{x_4\}\}$	AAAB

Note that Eq. (3.9), in which N denotes the population size, and Eq. (3.11) are equal. This correspondence between the two approaches evidences the conflict in

[1]The original formula in Steorts et al. (2016) has the factor $1/N^N$ rather than $1/N^S$, where S is the number of records.

SMERED with the definition of N because it was defined as "the total number of observed individuals from the population" Steorts et al. (2016, p. 1662). Despite Section 6 in Steorts et al. (2016) discusses the role of the population size, it does not clarify the topic. The concept of population remains unclear, and the authors recommend the topic for future research.

3.5 Summary

To sum up, the approaches proposed by Wright et al. (2009) and Steorts et al. (2016) describe a Bayesian model for linking observations that refer to the same entity. This model allows inferences about the real sample size. The full conditional densities of $\mathcal{G}$ and X are critical points in Wright et al. (2009). This chapter discussed and clarified some issues about these. The choice of either the indicator matrix X or the vector of indices y for modelling the connection between observed and true genotypes may seem a minor problem, but this chapter determined they have different models. For the indicator matrix, $f(X|N) = (1/N)^S$, while for the vector of indices, $f(y|N) = \frac{N!}{(N-n)!N^S}$. Also, the concept of population size is a problem that needs to be addressed in the case of Steorts et al. (2016). The authors started a discussion regarding this issue which does not solve it but suggests a strong similarity with the model proposed by Wright et al..

The two approaches were proposed in different fields, mark-recapture and record linkage. This discrepancy implies that the model by Steorts et al. can provide posterior matching probabilities between co-referent records. From the descriptions of the algorithms, there are clear differences for updating the indices that link observations with latent individuals, which in SMERED occurs by using a Metropolis step, and in GENUAD by a Gibbs step. One of the strengths of GENUAD is the fact that the population size can be estimated. In this sense, SMERED (or at least a modification) could be applied for updating (G, X) in GENUAD, which refers to the individuals in the population that are observed in the sample. The unobserved genotypes are sampled as in GENUAD, by using a categorical distribution.

Finally, the compatibility between observed and latent genotypes considered in GENUAD does not allow to use the SMERED algorithm since some inconsistencies related to the support of the joint density of $\mathcal{G}$ and X.

Chapter 4

Convergence of the Markov Chains

The previous chapter presented the GENUAD and SMERED algorithms to sample from the posterior distribution in Eq. (3.10). This chapter studies the convergence properties of the Markov chains they generate. Irreducibility, aperiodicity, ergodicity, and recurrence are some of the crucial concepts defined in Chapter 2 for proving the existence and uniqueness of the invariant distribution of a Markov chain. Moreover, satisfying Theorem 2.1.1 is a significant result to ensure the convergence of ergodic Markov chains with finite state space.

A Gibbs sampler may generate a reducible Markov chain, as illustrated in Example 2.2.1. Because the GENUAD algorithm is a Gibbs sampler, there is no guarantee of the irreducibility of the generated Markov chain. Thus, Section 4.1 examines the irreducibility of this chain. On the other hand, the convergence of the Markov chain generated by SMERED is determined in Section 4.2. In this case, the keyword is reversibility, which is a sufficient but not necessary condition for ensuring the existence of the stationary distribution.

4.1 GENUAD convergence

Section 3.2 introduced the GENUAD algorithm as a Gibbs sampler. The convergence of the corresponding Markov chain is studied by using the results in Section 2.2.1. As suggested by Theorem 2.2.2 a Markov chain generated by a Gibbs sampler may be reducible if the positivity condition does not hold. This reducibility prevents the ergodicity of the chain, which results in its non-convergence. Thus, irreducibility is a critical issue to be resolved in the case of GENUAD algorithm.

Definition 2.2.3 presented the positivity condition as a feature of the support of the joint density to be expressed as the Cartesian product of the support of the full conditional densities. According to Theorem 2.2.2, this condition is sufficient for concluding the irreducibility of the Markov chain. Considering the bivariate case, the positivity condition holds if $\text{supp}(f_{X,Y}) = \text{supp}(f_X) \times \text{supp}(f_Y)$, where f_X and f_Y are marginal densities and $f_{X,Y}$ their joint density. Figure 4.1 shows two joint densities with different support. A chain exploring the support in Figure 4.1(a) can move to any point with

positive probability. All the points can be reached by the chain starting at any point, which is exactly the definition of an irreducible chain. In contrast, the support of a joint density as in Figure 4.1(b) is an example in which the chain is trapped at subsets of the support.

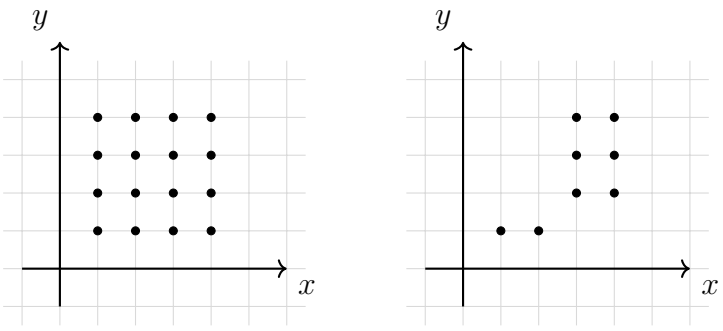


Figure 4.1: The support of the joint density in (a) is connected, while the support in (b) is not connected.

Although positivity is a sufficient condition for concluding irreducibility, it is not necessary. Figure 4.2 shows the support of a joint density which is not the Cartesian product of the marginal densities, which implies that the positivity condition is not satisfied. However, the Markov chain could still be irreducible. Unlike the support in Figure 4.1(b), the red point in Figure 4.2 connects the subsets where the chain gets stuck. A chain proceeds between the two regions by using that connecting point. The existence of those points for connecting isolated regions of the support allows the chain to move freely through the state space without being trapped in a specific region. Thus, positivity is a sufficient but not necessary condition for irreducibility.

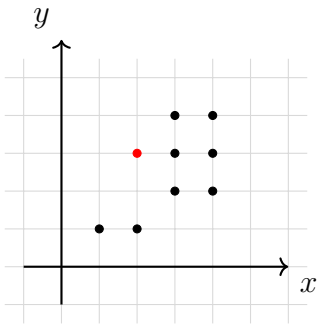


Figure 4.2: An example of a bivariate support where positivity does not hold.

Considering the model by Wright et al. (2009), and giving fixed values to the parameters N, γ and p (refer to page 9 for the definition of these quantities.), it can be shown that the support of the joint density of $\mathcal{G}$ and X is not the Cartesian product of the support of the marginal densities. In fact,

$$\text{supp}(f_{\mathcal{G},X}) \subset \text{supp}(f_{\mathcal{G}}) \times \text{supp}(f_X). \tag{4.1}$$

Thus, the joint density of $\mathcal{G}$ and X does not satisfy the positivity condition. The following definition characterises the support of $f_{\mathcal{G},X}$. Definition A.1.3 of compatibility is required.

Definition 4.1.1. Let g^{obs} be the observed genotypes in a sample with size S, $\mathcal{G}$ the true genotypes in the population, and X the indicator matrices for the membership of an individual in the sample. The support of the joint density of $\mathcal{G}$ and X in Eq. (1.2) is defined as

$$\text{supp}(f_{\mathcal{G},X}) = \{(\mathcal{G}, X) : g_{il}^{\text{obs}} \text{ is compatible with } \mathcal{G}_{y_i l} \ \forall i = 1, \ldots, S \text{ and } l = 1, \ldots, L\}$$

where y is a S-dimensional vector with n unique membership indices, that is, y contains the same information as X, but only for those individuals that were observed in the sample.

The following example illustrates why the inclusion in the opposite direction in Eq. (4.1) does not hold and how two different states in $\text{supp}(f_{\mathcal{G},X})$ are connected.

Example 4.1.1. Let $g^{\text{obs}} = \begin{pmatrix} 1,1 \\ 2,2 \end{pmatrix}$ be the observed genotypes at a single locus with two alleles (i.e. $S = 2$, $L = 1$, $m = 2$). Because there are two alleles, the three possible genotypes at the locus are: $\{(1,1),(1,2),(2,2)\}$. The order of the indicator matrix X is $N \times S$, which means that the number of possible matrices X is N^S. Assuming that $N = 3$, Table 4.1 shows the nine elements of $\text{supp}(f_X)$, which are 3×2 matrices. Each column of X is represented as a row of three elements in the table. Table 4.2 shows the $N^3 = 27$ elements of $\text{supp}(f_{\mathcal{G}})$. For each row (individual) of $\mathcal{G}$, there are three possible genotypes, which are represented by pairs of numbers.

Table 4.1: The elements in $\text{supp}(f_X)$.

j	Column 1			Column 2		
1	1	0	0	1	0	0
2	0	1	0	1	0	0
3	0	0	1	1	0	0
4	1	0	0	0	1	0
5	0	1	0	0	1	0
6	0	0	1	0	1	0
7	1	0	0	0	0	1
8	0	1	0	0	0	1
9	0	0	1	0	0	1

For example, if $j = 2$ and $i = 11$ in Tables 4.1 and 4.2, respectively, the pair $(\mathcal{G}^{11}, X^2)$ is given by

$$\mathcal{G}^{11} = \begin{pmatrix} 1,2 \\ 1,1 \\ 1,2 \end{pmatrix} \quad \text{and} \quad X^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Table 4.2: The elements in $\text{supp}(f_{\mathcal{G}})$.

i	Individual 1	Individual 2	Individual 3
1	1 1	1 1	1 1
2	1 2	1 1	1 1
3	2 2	1 1	1 1
4	1 1	1 2	1 1
5	1 2	1 2	1 1
6	2 2	1 2	1 1
7	1 1	2 2	1 1
8	1 2	2 2	1 1
9	2 2	2 2	1 1
10	1 1	1 1	1 2
11	1 2	1 1	1 2
12	2 2	1 1	1 2
13	1 1	1 2	1 2
14	1 2	1 2	1 2
15	2 2	1 2	1 2
16	1 1	2 2	1 2
17	1 2	2 2	1 2
18	2 2	2 2	1 2
19	1 1	1 1	2 2
20	1 2	1 1	2 2
21	2 2	1 1	2 2
22	1 1	1 2	2 2
23	1 2	1 2	2 2
24	2 2	1 2	2 2
25	1 1	2 2	2 2
26	1 2	2 2	2 2
27	2 2	2 2	2 2

From g^{obs}, the state space of g is given by

$$\mathcal{X}_g = \left\{ \begin{pmatrix} 1,1 \\ 1,2 \end{pmatrix}, \begin{pmatrix} 1,1 \\ 2,2 \end{pmatrix}, \begin{pmatrix} 1,2 \\ 1,2 \end{pmatrix}, \begin{pmatrix} 1,2 \\ 2,2 \end{pmatrix} \right\}. \tag{4.2}$$

Each matrix $g^i \in \mathcal{X}_g$, for $i \in \{1,2,3,4\}$, is determined by a pair $(\mathcal{G}, X) \in \text{supp}(f_{\mathcal{G},X})$ and g^{obs}, according to Definition 3.1.1. However, a matrix g^i may be generated from different pairs $(\mathcal{G}, X)$. For example, $(\mathcal{G}^{11}, X^2)$ as above gives g^1. But g^1 is also generated by $(\mathcal{G}^5, X^3)$.

Figure 4.3 shows $\text{supp}(f_{\mathcal{G},X})$ for this example. No matter the colour, a dot represents a pair $(\mathcal{G}, X) \in \text{supp}(f_{\mathcal{G},X})$. Because each g^i may be represented by different pairs $(\mathcal{G}, X)$, there are multiple dots with the same colour. Following the same order as in Eq. (4.2), the colours were assigned according to the set $\{\text{black}, \text{green}, \text{red}, \text{blue}\}$. Note that there are elements in the Cartesian product of the supports of the marginal

densities which are not necessarily in the support of the joint density. For example, consider $\mathcal{G}^{16}$ and X^1 which, according to Tables 4.2 and 4.1, respectively, are given by

$$\mathcal{G}^{16} = \begin{pmatrix} 1,1 \\ 2,2 \\ 1,2 \end{pmatrix} \quad \text{and} \quad X^1 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}.$$

$\mathcal{G}^{16} \in \text{supp}(f_{\mathcal{G}})$ because it can produce the first, second and fourth elements in $\mathcal{X}_g$ (i.e. $f_{\mathcal{G}}(\mathcal{G}^{16}) > 0$); and $X^1 \in \text{supp}(f_X)$ because of the third element in $\mathcal{X}_g$ (i.e. $f_X(X^1) > 0$). However, $(\mathcal{G}^{16}, X^1) \notin \text{supp}(f_{\mathcal{G},X})$ because $\begin{pmatrix} 1,1 \\ 1,1 \end{pmatrix} \notin \mathcal{X}_g$. This is because the second observed genotype $2,2$ in g^{obs} is not compatible with the first true genotype $1,1$ in $\mathcal{G}$.

Although g^3 may be generated from the pair $(\mathcal{G}^{11}, X^7)$ according to Definition 3.1.1, $(\mathcal{G}^{11}, X^7) \notin \text{supp}(f_{\mathcal{G},X})$ because identical true genotypes in the sample refer to the same observed individual. That is, twins are not allowed in the sample. This assumption was discussed on page 41 after numeral 4.

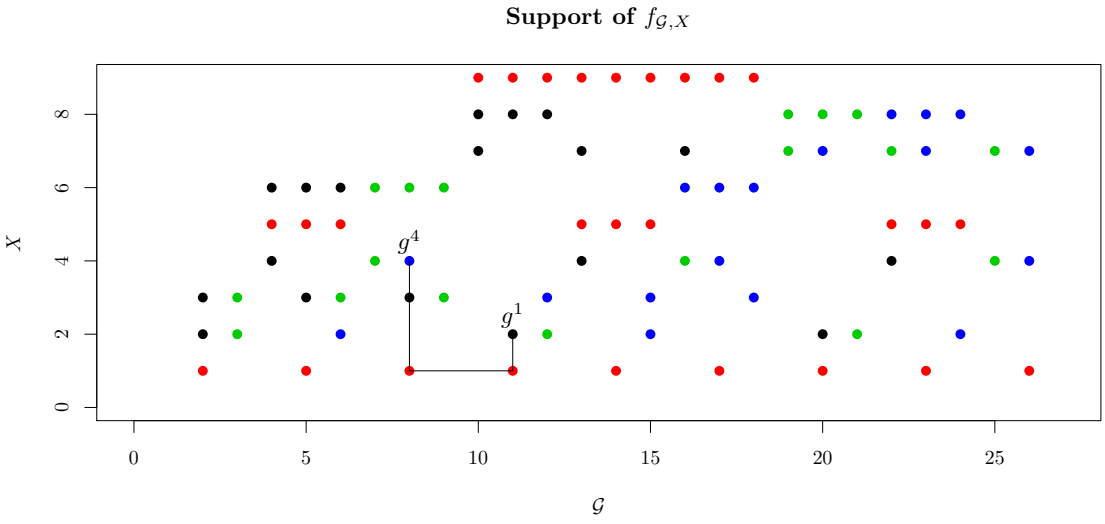


Figure 4.3: A dot represents a pair $(\mathcal{G}, X) \in \text{supp}(f_{\mathcal{G},X})$, and it corresponds to a state $g \in \mathcal{X}_g$. Each element of $\mathcal{X}_g = \{g^1, g^2, g^3, g^4\}$ in Eq. (4.2) is represented with a colour in $\{\text{black}, \text{green}, \text{red}, \text{blue}\}$, respectively. The path shows how the states in $\text{supp}(f_{\mathcal{G},X})$, associated with g^1 and g^4, connect.

The states $(\mathcal{G}^{11}, X^2)$ and $(\mathcal{G}^8, X^4)$ are considered to show how the states in $\text{supp}(f_{\mathcal{G},X})$ connect. These states generate the following g matrices.

$$\mathcal{G}^{11} = \begin{pmatrix} 1,2 \\ 1,1 \\ 1,2 \end{pmatrix} \quad \text{and} \quad X^2 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \Rightarrow \quad g^1 = \begin{pmatrix} 1,1 \\ 1,2 \end{pmatrix},$$

$$\mathcal{G}^8 = \begin{pmatrix} 1,2 \\ 2,2 \\ 1,1 \end{pmatrix} \quad \text{and} \quad X^4 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \quad \Rightarrow \quad g^4 = \begin{pmatrix} 1,2 \\ 2,2 \end{pmatrix}.$$

The strategy adopted to show that $(\mathcal{G}^8, X^4)$ can be reached from $(\mathcal{G}^{11}, X^2)$ utilises the g matrices. In this case, g^1 and g^4 are compared row by row. Starting from g^1, all rows in which g^1 and g^4 differ are updated. They are different in the two rows. Each case is analysed as follows. The full conditional densities described in Section 3.2.2 are needed.

- The first row in g^1 will change to be $1,2$, which is the first row in g^4. Conveniently, because $1,2$ is an element $\mathcal{G}^{11}$, X^2 is updated given this current value of $\mathcal{G}$. With positive probability $X = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$ can be obtained. The current pair is $(\mathcal{G}^{11}, X^1) \in \text{supp}(f_{\mathcal{G},X})$ which generates $g^3 = \begin{pmatrix} 1,2 \\ 1,2 \end{pmatrix}$. The first row of g^1 becomes 1,2.
- The second row in g^3 will change to be $2,2$, as the second row in g^4. Because $2,2$ is not an element $\mathcal{G}^{11}$, it is updated given X^1. With positive probability $\mathcal{G} = \begin{pmatrix} 1,2 \\ 2,2 \\ 1,1 \end{pmatrix}$ can be obtained. The current pair is $(\mathcal{G}^8, X^1) \in \text{supp}(f_{\mathcal{G},X})$ which also generates g^3. Given $\mathcal{G}^8$, X^1 is updated and with positive probability $X = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$ can be obtained. The current pair $(\mathcal{G}^8, X^4) \in \text{supp}(f_{\mathcal{G},X})$ generates target state g^4.

When starting at g^1, the procedure above shows how to reach g^4 by updating each row in which they differ.

□

Example 4.1.1 showed that $f_{\mathcal{G},X}$ does not uphold the positivity condition. Consequently, the irreducibility of the Markov chain generated by GENUAD cannot be guaranteed by using Theorem 2.2.2. Also, the example introduces the alternative strategy to be used. The solution proposed is simple and involved the definition of communicability presented in Section 2.1. That is, if the chain starts at an arbitrary and initial state, with positive probability, it will reach any other state in a finite number of steps. The next theorem generalizes this strategy for connecting the elements in $\text{supp}(f_{\mathcal{G},X})$ through the g matrices. The proof requires the introduction of some notation. Following Wright et al. (2009), every component in a pair $(\mathcal{G}, X)$ can be partitioned as $\mathcal{G} = (G^{\text{obs}}, G^{\text{mis}})'$ where G^{obs} denotes the true genotypes of the individuals that were observed in the sample, and G^{mis} for those that were not observed. The indicator matrix X has $N - n$ rows of zeros for indicating the absence of the true genotypes in G^{mis} in the sample.

Theorem 4.1.1. Let $\mathcal{S} = \text{supp}(f_{\mathcal{G},X})$, as defined in Definition 4.1.1, for a known matrix g^{obs}. If $(\mathcal{G}^{(0)}, X^{(0)}) \in \mathcal{S}$ is some fixed initial state of the chain, then for any $(\mathcal{G}^{(k)}, X^{(k)}) \in \mathcal{S}$ there exists a sequence of states in $\mathcal{S}$,

$$(\mathcal{G}^{(0)}, X^{(1)}), (\mathcal{G}^{(1)}, X^{(1)})(\mathcal{G}^{(1)}, X^{(2)}), (\mathcal{G}^{(2)}, X^{(2)}) \ldots, (\mathcal{G}^{(k-1)}, X^{(k)}), (\mathcal{G}^{(k)}, X^{(k)}),$$

such that $(\mathcal{G}^{(0)}, X^{(0)})$ and $(\mathcal{G}^{(k)}, X^{(k)})$ communicate.

Proof. Let $(\mathcal{G}^{(0)}, X^{(0)}) \in \mathcal{S}$ a fixed initial state and $(\mathcal{G}^{(k)}, X^{(k)}) \in \mathcal{S}$ any other state. Each pair $(\mathcal{G}, X) \in \mathcal{S}$ determines a matrix g which contains the true genotypes of the individuals in the sample. In this case, the states above generate $g^{(0)}$ and $g^{(k)}$, with S rows but $n^{(0)}$ and $n^{(k)}$ unique rows, respectively. The adopted strategy consists of comparing $g^{(0)}$ and $g^{(k)}$ row by row. There are two cases, either $g^{(0)} \neq g^{(k)}$ or $g^{(0)} = g^{(k)}$.

If $g^{(0)} \neq g^{(k)}$, the rows in which these matrices differ will define the sequence. Let $\mathcal{I}$ denote the set of indices for which $g^{(0)}$ and $g^{(k)}$ are different, that is, $\mathcal{I} = \{i : g_i^{(0)} \neq g_i^{(k)}, i = 1, \ldots, S\}$. For $i \in \mathcal{I}$, one of the following two cases holds:

i. $g_i^{(k)} = g_j^{(0)}$ for some $j = 1, \ldots, S$ and $j \neq i$.

ii. $g_i^{(k)} \neq g_{j:}^{(0)}$ for all $j = 1, \ldots, S$ and $j \neq i$.

If (i.), then $g_i^{(k)}$ is a row in the observable partition G^{obs} of $\mathcal{G}^{(0)}$, say row r_i with $r_i \leq n^{(0)}$. The number of elements in any G^{obs}, n, must satisfy $n < N$ to ensure that the chain does not get stuck. Given $\mathcal{G}^{(0)}$, $X^{(0)}$ is updated such that the entry in column i and row r_i, which is currently 0, becomes equal to 1, and the entry with 1 (whose index is known to be different to r_i because $i \in \mathcal{I}$) becomes 0. This leads to a new matrix $X^{(1)}$ such that $(\mathcal{G}^{(0)}, X^{(1)}) \in \mathcal{S}$. Given $X^{(1)}$, $\mathcal{G}^{(0)}$ is updated. With positive probability, a new matrix $\mathcal{G}^{(1)}$ is obtained such that its r_ith row is equal to $g_i^{(k)}$, and $(\mathcal{G}^{(1)}, X^{(1)}) \in \mathcal{S}$.

If (ii.), then $g_i^{(k)}$ cannot be found in the observable partition of $\mathcal{G}^{(0)}$, and there is no guarantee it will be in G^{mis}. Given the current $X^{(0)}$, $\mathcal{G}^{(0)}$ is updated and with positive probability a matrix $\mathcal{G}^{(1)}$ can be obtained such that $g_i^{(k)}$ is its r_ith row, for $r_i > n^{(0)}$ (i.e. $g_i^{(k)}$ is a row in the relevant G^{mis}), and $(\mathcal{G}^{(1)}, X^{(0)}) \in \mathcal{S}$. Given $\mathcal{G}^{(1)}$, $X^{(0)}$ is updated such that the non-zero entry in column i, currently in a row different to r_i, shifts to it. Then, a new indicator matrix $X^{(1)}$ is obtained, for which $(\mathcal{G}^{(1)}, X^{(1)}) \in \mathcal{S}$.

In both cases, a state $(\mathcal{G}^{(1)}, X^{(1)}) \in \mathcal{S}$ is generated, which determines a matrix $g^{(1)}$ such that $g_i^{(1)} = g_i^{(k)}$ (i.e. $g^{(1)}$ and $g^{(k)}$ coincide in the ith row). Following the same argument for every $i \in \mathcal{I}$, there exists a state g^i resulting from a pair $(\mathcal{G}^i, X^i) \in \mathcal{S}$ such that $g_i^i = g_i^{(k)}$. Notice that $k = |\mathcal{I}|$. Therefore, a sequence of matrix $\{g^{(0)}, g^{(1)}, g^{(2)}, \ldots, g^{(k)}\}$ has been found, which allows moving to $g^{(k)}$ from $g^{(0)}$.

When the g matrices associated with $(\mathcal{G}^{(0)}, X^{(0)})$ and $(\mathcal{G}^{(k)}, X^{(k)})$ are equal, the construction of the sequence is straightforward. If $g^{(0)} = g^{(k)}$, the observable partitions G^{obs} of $\mathcal{G}^{(0)}$ and $\mathcal{G}^{(k)}$ are equal. Then, the $\mathcal{G}$ and X matrices in the sequence are

permutations of the rows in that observable partition. The G^{mis} partitions are the same with positive probability because those rows correspond to individuals never seen in the observed sample.

□

In Example 4.1.1, the sequence for connecting g^1 and g^4, represented by the path in Figure 4.3, is

$$(\mathcal{G}^{11}, X^2), (\mathcal{G}^{11}, X^1), (\mathcal{G}^8, X^1), (\mathcal{G}^8, X^4),$$

which in terms of the g matrices is equivalent to the sequence g^1, g^3, g^3, g^4. Note that there is not a unique sequence, as different states $(\mathcal{G}, X)$ may generate the same state g. Figure 4.4 shows a different sequence given by

$$(\mathcal{G}^{11}, X^2), (\mathcal{G}^{11}, X^1), (\mathcal{G}^5, X^1), (\mathcal{G}^5, X^6), (\mathcal{G}^8, X^6), (\mathcal{G}^8, X^4),$$

equivalent to $g^1, g^3, g^3, g^1, g^2, g^4$.

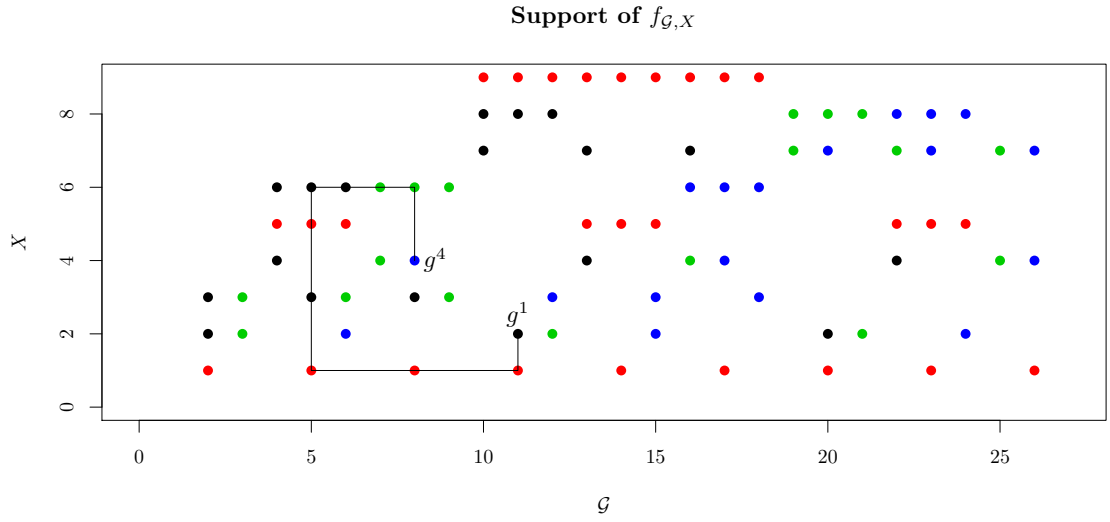


Figure 4.4: A different sequence for travelling from $(\mathcal{G}^{11}, X^2)$ to $(\mathcal{G}^8, X^4)$.

Several consequences follow from the theorem above. First, Theorem 4.1.1 and Lemma 2.2.1 lead to the conclusion that the full conditional densities $f_{\mathcal{G}|X}$ and $f_{X|\mathcal{G}}$ uniquely determine the joint density $f_{\mathcal{G},X}$. Second, the construction of the sequence in Theorem 4.1.1 shows how a target state $(\mathcal{G}^{(k)}, X^{(k)})$ can be reached from any initial state $(\mathcal{G}^{(0)}, X^{(0)})$. That is, any initial state $(\mathcal{G}^{(0)}, X^{(0)})$ communicates with any other state in $\mathcal{S}$, which implies that any pair of states in $\mathcal{S}$ communicate by using the initial state as an intermediate state. This property indicates that the Markov chain generated by GENUAD is irreducible. The following theorem and corollaries summarise these results.

Corollary 4.1.1. The existence of a unique joint density $f_{\mathcal{G},X}$ is determined by the full conditional densities$f_{\mathcal{G}|X}$ and $f_{X|\mathcal{G}}$.

Proof. Theorem 4.1.1 ensures the existence of a sequence $g^{(0)}, g^{(1)}, \ldots, g^{(k)}$ for communicating the initial state $g^{(0)}$ and any state $g^{(k)}$. The proof shows that each state $g^{(i)}$ results either from $(\mathcal{G}^{(i)}, X^{(i-1)})$ and $(\mathcal{G}^{(i)}, X^{(i)})$, or $(\mathcal{G}^{(i-1)}, X^{(i)})$ and $(\mathcal{G}^{(i)}, X^{(i)})$. In either case, the states differ in a single component. Besag's condition in Lemma 2.2.1 asserts that if there exists a finite sequence of states such that successive states differ only in a single component, then the full conditional densities determine the joint density of $\mathcal{G}$ and X. Thus, the result follows. □

Note that Besag's Lemma 2.2.1 is applicable despite Hobert et al. (1997) since the state space $\mathcal{S}$ corresponding to the Markov chain generated by GENUAD is discrete.

Corollary 4.1.2. The Markov chain generated by the GENUAD algorithm, as explained in Algorithm 1, is irreducible.

Aperiodicity is satisfied in the GENUAD algorithm because self-loops have positive probability. That is, starting in an initial state $(\mathcal{G}^{(0)}, X^{(0)})$, with positive probability the Markov chain may generate a state $(\mathcal{G}^{(0)}, X^{(1)})$ such that $X^{(0)} = X^{(1)}$, which is equivalent to saying that the period of this state is 1. As explained in Definition 2.1.9, irreducibility allows to conclude the aperiodicity of the chain. Then, the GENUAD Markov chain is aperiodic.

Corollary 4.1.3. The Markov chain generated by the GENUAD algorithm is ergodic.

Since the GENUAD Markov chain is aperiodic and irreducible, the proof follows from Definition 2.1.11. All the conditions in Theorem 2.1.1 have been met to conclude that the Markov chain generated by the GENUAD algorithm converges to its invariant distribution.

4.2 SMERED convergence

This section discusses potential problems with the convergence of the SMERED algorithm outlined in Algorithm 2. The following two problems were identified:

(P1) The strategy for updating G in step 9 only allows sampling from the observations.

(P2) The Metropolis ratio defined in step 10 was not correctly defined.

Regarding (P1), Example 3.3.1 illustrates the process for updating G when splitting two records, which only considers a subset of observations (see numeral vi. in the list of steps). Although the algorithm includes a resampling step (12), after deciding about the acceptance/rejection of the proposal, it is possible that this strategy leads to a reducible chain under other kind of corruption process. The genotyping error considered in GENUAD (allelic dropout) is an example of this situation. Example 3.4.1 shows a pair $(\mathcal{G}^*, X^*)$ which does not belong to the support of the joint density $f(\mathcal{G}, X)$ (i.e. it has probability zero). Thus, limiting the sampling in step 9 to the observations set may results in a reducible chain depending on the error involved into the data.

In general, even if the chain is irreducible, it is not guaranteed to generate a Markov chain with the correct limiting distribution.

While irreducibility is the issue in (P1), reversibility is the concern in (P2). Robert and Casella (2004) state that the existence of the invariant distribution of a Markov chain generated by a Metropolis algorithm follows by construction. The reason is that the Metropolis ratio is defined such that the transition kernel satisfies the reversibility condition, as clearly explained by Chib and Greenberg (1995). For the particular case of the SMERED algorithm, it seems that the proposal distribution for sampling a pair (G, y) was assumed to be symmetric, since the relevant ratio does not appear in the expression. That is, the term $q(x^*, x^{t-1})/q(x^{t-1}, x^*)$, where $x = (G, y)$, in Eq. (2.10) is equal to 1.0. However, the proposal distribution is not symmetric, as shown below. Thus, this inaccuracy casts serious doubt on the existence of the invariant distribution.

To illustrate that the proposal density is not symmetric, consider Example 3.3.1. A split was proposed, denoted by (G^*, y^*), after randomly choosing record 2 from file 1, and record 3 from file 3. The problem is that the reverse move is impossible. There is no way to go from (G^*, y^*) to (G, y) due to the reverse move is equivalent to merging the records. As previously explained, the construction of a set of records C currently assigned to individuals 2 and 5 is required for constructing the set from which the new value for G will be sampled. In this case, $C = \{(SC, 70, F), (SC, 37, F), (SC, 72, F)\}$. This set does not contain the value $(SC, 73, F)$, which is indexed as 2 in G. Therefore, it is crucial to improve the scheme for the proposals outlined by SMERED, more specifically step 9 in Algorithm 2.

Because reversibility is not satisfied, the existence of a unique stationary distribution cannot be ensured. A simple, but clumsy, solution would be to introduce the ratio corresponding to the proposal density. That is, $q(x^*, x^{t-1})/q(x^{t-1}, x^*)$ where $x = (G, y)$. This ratio, however, would lead to frequent rejections which correspond to the states that cannot be reached. In other words, ensuring reversibility leads to an inefficient algorithm, one unable to freely explore the state space.

Although reversibility is a sufficient condition for convergence, it is not necessary, as shown by Example 2.1.1. Only a counterexample could prove that, in effect, the Markov chain generated by the SMERED algorithm does not converge to its invariant distribution. With that aim, the badgers genotypes will be used. The idea is using Algorithm 2, in an attempt to reproduce the posterior joint distribution of G and X, given in Eq. (3.10). As discussed in Section 3.4.2, the models in Wright et al. (2009) and Steorts et al. (2016) contain major differences which make them incomparable. Therefore, to guarantee that the models are comparable, they must be set under the same conditions.

Wright et al. (2009), contrary to Steorts et al. (2016), specified a particular corruption process of the data which is due to allelic dropout. For this reason, the example presented here does not consider the genotyping error introduced by allelic dropout. Instead, the data here are assumed to have all types of contamination rather than

allelic dropout only. That is, any corrupted genotype may be co-referent to any other without restrictions.

For instance, under the corruption process in GENUAD, a true homozygote AA is only linked to either AA or AB, where B $\neq$ A. However, under the new corruption process described here, it can be linked to any combination of two alleles. The cases are AA, AX, XX, and XY where X,Y $\neq$ A. For loci with two alleles, the heterozygote XY would not be a case for AA. If p denotes the probability of corruption of an allele, m the number of alleles at the locus and the corruption is independent among alleles, then the probabilities of these possible cases are as follows.

For the first case, because the two alleles are not corrupted,

$$\Pr(g^{\text{obs}} = \text{AA}|g = \text{AA}) \;=\; (1-p)^2.$$

For the second case,

$$\Pr(g^{\text{obs}} = \text{AX}|g = \text{AA}) \;=\; \frac{2p(1-p)}{m-1}$$

because there is corruption in only one allele, X is one of $m-1$ alleles distinct to A, and AX = XA. For the third case,

$$\Pr(g^{\text{obs}} = \text{XX}|g = \text{AA}) \;=\; \left(\frac{p}{m-1}\right)^2$$

because both alleles are corrupted and X is one of $m-1$ alleles distinct to A. For the last case,

$$\Pr(g^{\text{obs}} = \text{XY}|g = \text{AA}) \;=\; 2\left(\frac{p}{m-1}\right)^2$$

because both alleles are corrupted, X and Y are one of $m-1$ alleles distinct to A, and XY = YX.

For the case in which the true genotype is heterozygous, say AB, the possible outcomes for the observed genotypes are AB, AA, BB, AX, BX, XX, and XY, where $\{\text{X,Y}\} \cap \{\text{A,B}\} = \emptyset$. Their probabilities can be found following a similar process. The probabilities of the new corruption process are summarised below, where p is the probability of corruption of an allele, and m the number of alleles at the locus.

For g = AA,

$$\Pr(g^{\text{obs}}|g,p) = \begin{cases} (1-p)^2 & \text{if } g^{\text{obs}} = \text{AA}, \\ \left(\dfrac{p}{m-1}\right)^2 & \text{if } g^{\text{obs}} = \text{XX with A} \neq \text{X}, \\ 2\left(\dfrac{p}{m-1}\right)^2 & \text{if } g^{\text{obs}} = \text{XY with A} \neq \text{X and A} \neq \text{Y}, \\ \dfrac{2p(1-p)}{m-1} & \text{otherwise.} \end{cases} \tag{4.3}$$

For $g = \text{AB}$,

$$\Pr(g^{\text{obs}}|g,p) = \begin{cases} (1-p)^2 + \left(\dfrac{p}{m-1}\right)^2 & \text{if } g^{\text{obs}} = \text{AB}, \\ \left(\dfrac{p}{m-1}\right)^2 & \text{if } g^{\text{obs}} = \text{XX with X} \neq \text{A and X} \neq \text{B}, \\ \dfrac{p(1-p)}{m-1} & \text{if } g^{\text{obs}} = \text{AA or } g^{\text{obs}} = \text{BB}, \\ 2\left(\dfrac{p}{m-1}\right)^2 & \text{if } g^{\text{obs}} = \text{XY with } \{\text{X,Y}\} \cap \{\text{A,B}\} = \emptyset, \\ \dfrac{p(1-p)}{m-1} + \left(\dfrac{p}{m-1}\right)^2 & \text{otherwise.} \end{cases} \tag{4.4}$$

To ensure that the probabilities in Eqs. (4.3) and (4.4) have been correctly specified, the sum to 1.0 for each will be examined. The idea is to count how many cases hold the condition of g^{obs}. Tables 4.3 and 4.4 show the sums for both true homozygote and true heterozygote cases, respectively.

Table 4.3: Counting cases for g^{obs} when $g = \text{AA}$, and $\text{X} \neq \text{A}$.

g^{obs}	Counting	Counting$\cdot \Pr(g^{\text{obs}}\|g,p)$
AA	1	$(1-p)^2$
XX	$m-1$	$p^2/(m-1)$
XY	$(m-1)(m-2)/2$	$(m-2)p^2/(m-1)$
AX	$m-1$	$2p(1-p)$

Table 4.4: Counting cases for g^{obs} when $g = \text{AB}$, and $\text{X,Y} \notin \{\text{A,B}\}$

g^{obs}	Counting	Counting$\cdot \Pr(g^{\text{obs}}\|g,p)$
AB	1	$(1-p)^2 + \left(\dfrac{p}{m-1}\right)^2$
XX	$m-2$	$(m-2)\left(\dfrac{p}{m-1}\right)^2$
AA or BB	2	$\dfrac{2p(1-p)}{m-1}$
XY	$\dfrac{(m-3)(m-2)}{2}$	$(m-3)(m-2)\left(\dfrac{p}{m-1}\right)^2$
AX or BX	$2(m-2)$	$2(m-2)\left[\dfrac{p(1-p)}{m-1} + \left(\dfrac{p}{m-1}\right)^2\right]$

Summing the third column in Table 4.3,

$$\begin{aligned}\sum_{g^{\text{obs}}} \Pr(g^{\text{obs}}|g=\text{AA},p) &= 1-2p+p^2+\frac{p^2}{m-1}+\frac{m-2}{m-1}p^2+2p-2p^2 \\ &= 1+\left(\frac{1}{m-1}+\frac{m-2}{m-1}-1\right)p^2 \\ &= 1.\end{aligned}$$

Similarly, summing the third column in Table 4.4,

$$\begin{aligned}\sum_{g^{\text{obs}}} \Pr(g^{\text{obs}}|g=\text{AB},p) &= (1-p)^2+(m-1)^2\left(\frac{p}{m-1}\right)^2+2(m-1)\frac{p(1-p)}{m-1} \\ &= 1-2p+p^2+p^2+2p-2p^2 \\ &= 1.\end{aligned}$$

Thus, the probabilities associated with the new corruption process have been defined, to ensure the possible events for g^{obs} are both exhaustive and mutually exclusive. The probabilities $\Pr(g^{\text{obs}}|g,p)$ above modify those in Section A.2.1 to make the algorithms comparable.

Now both models have a similar corruption process, which is crucial for a valid and fair comparison. Under these aligned conditions, the data as considered by Wright et al. (2009) can be modelled by implementing the SMERED algorithm. The following example illustrates the performance of SMERED in such a situation. The purpose of the small dataset is to take advantage of the small size of the state space, which allows the inclusion of the analytical joint distribution of interest. In this way, comparisons between the simulated and the exact distributions are achievable.

Example 4.2.1. Consider a sample with $S=2$ observed genotypes at a single locus with $m=2$ alleles, as follows.

$$g^{\text{obs}} = \begin{pmatrix} 1,1 \\ 1,2 \end{pmatrix}.$$

According to the new corruption process, if an observed genotype is corrupted, then it could be any one of these true genotypes: $\{(1,1),(1,2),(2,2)\}$. If $\mathcal{X}_g$ denotes the state space of g (the true genotypes that were observed in the sample), this state space has, then, $3^2=9$ elements.

$$\mathcal{X}_g = \left\{ \begin{pmatrix} 1,1 \\ 1,1 \end{pmatrix}, \begin{pmatrix} 1,1 \\ 1,2 \end{pmatrix}, \begin{pmatrix} 1,1 \\ 2,2 \end{pmatrix}, \ldots, \begin{pmatrix} 2,2 \\ 1,2 \end{pmatrix}, \begin{pmatrix} 2,2 \\ 2,2 \end{pmatrix} \right\}.$$

Defining $N=3, \gamma=(0.5,0.25,0.25)'$ and $p=0.25$, the SMERED algorithm is implemented to draw samples from the posterior distribution in Eq. (3.10). The density function $f(g^{\text{obs}}|G,y,p)$ has been defined above, $f(G|N,\gamma)$ is determined by γ, and

$f(y|N)$ is defined in Eq. (3.9). The aim with this small example is to compare the exact posterior distribution to that simulated by SMERED. The following procedure shows how to find the exact probability for the first element in the set above. That is, $g = \begin{pmatrix} 1,1 \\ 1,1 \end{pmatrix}$, which means that $G = \begin{pmatrix} 1,1 \end{pmatrix}$.

$$\begin{aligned} f(g^{\text{obs}}|G,y,p) &= (1-p)^2 \cdot \frac{2p(1-p)}{m-1} = 0.2109375 \\ f(y|N) &= \frac{N!}{(N-n)!}\left(\frac{1}{N}\right)^S = \frac{3!}{2!}\left(\frac{1}{3}\right)^2 = \frac{1}{3} \\ f(G|N,\gamma) &= \gamma_{1,1} = 0.5 \end{aligned}$$

As follows, the posterior probability of g given $g^{\text{obs}}, \gamma, N$, and p is proportional to the product of these three terms, which is equal to 0.03515625. Repeating this process for all nine states, and finding the normalizing constant, the exact probabilities are given by,

$$(0.33123, 0.2761, 0.1656, 0.0552, 0.0920, 0.0276, 0.0184, 0.0153, 0.0184).$$

These probabilities are shown in Figure 4.5 using the red triangles. The black points correspond to a simulation using 100 000 iterations of the SMERED algorithm. The error bars were constructed by using multiple chains, and using the between-chain variance.

Following Gelman et al. (2004), suppose M chains, each with length T. The mth chain is a sequence $\{\theta_{m1}, \ldots, \theta_{mT}\}$ where θ_{mi} is some integer in $\{1, \ldots, 9\}$, for labelling the states in $\mathcal{X}_g$. Let $\hat{h}_i$ denote the Monte Carlo estimate of the proportion of visits to the state g_i, $i = 1, \ldots, 9$. The aim is to estimate the variance of each $\hat{h}_i$.

Let $\hat{h}_{im}$ be the estimate for chain m. The estimated variance of the Monte Carlo estimate is

$$\text{var}(\hat{h}_i) = \frac{1}{M-1}\sum_{j=1}^{M}(\hat{h}_{ij} - \hat{h}_i^*)^2 \text{ where } \hat{h}_i^* = \frac{1}{M}\sum_{j=1}^{M}\hat{h}_{ij}$$

The point estimates can be assumed as normally distributed if M is large (using the central limit theorem). Figure 4.5 shows the estimated 95% confidence intervals (error bars) for the true proportions of visits to each state in $\mathcal{X}_g$ for $M = 45$ chains with the same length of $T = 100000$. The chains were initialised in different states of $\mathcal{X}_g$. This experiment strongly indicates that the Markov chain generated by the SMERED algorithm does not converge to the correct stationary distribution.

□

Example 4.2.1 shows evidence that the Markov chain generated by the SMERED algorithm may not converge to the distribution it was designed to simulate. This failure is a consequence of the problems (P1) and (P2), which are reducibility and non-reversibility problems, respectively. These problems are closely connected to the dimension of G, as a consequence of the split-merge operations applied in SMERED.

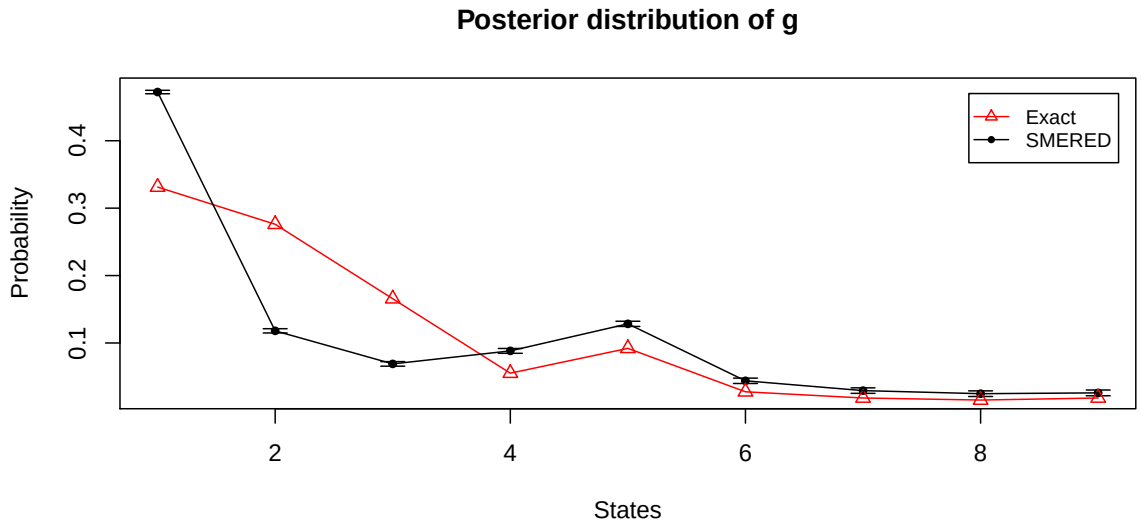


Figure 4.5: The red triangles indicate the exact invariant distribution of g. The black points indicate the proportion of visits of the chain simulated by SMERED (under the new corruption process) to each of the nine states in $\mathcal{X}_g$. Also, the estimated (error bars) 95% confidence intervals of the proportions of visits from a sample of $M = 45$ chains are shown.

They are based on the procedure considered by Jain and Neal (2004), previously proposed by Green and Richardson (2001), in the context of Dirichlet process mixture models. The idea is to enhance a M-H algorithm regarding its efficiency for moving through the space state, and this is done by splitting or merging the mixture components. The approach in Jain and Neal (2004) takes full advantage of the conjugacy in the model to analytically integrate over the mixing proportions and component-specific parameters, leaving only the latent indicator variables. These indicators are then updated through splitting and merging steps.

In the context of SMERED, the latent indicator variables mentioned above correspond to the values in the linkage structure y, and the parameters for each component correspond to G. Indeed, the linkage structure is updated by SMERED using split-merge operations, but G is jointly updated, instead of marginalised. The problem is that there are no such conjugacy properties in the SMERED model for integrating away G. Instead, the process for updating G and y causes the dimension of G to increase (when splitting) or decrease (when merging) by one unit at each iteration of the algorithm. Thus, if the change in the dimensionality of G is taken into account, then G and y could potentially be jointly and correctly updated.

To illustrate how G changes dimension when y is updated, consider Example 3.3.1. Updating the linkage structure is equivalent to creating a new index for identifying the individuals, which is not added to the existing indexes, and is a replacement instead. That is, the dimension of y remains unmodified. However, a new row (genotype) in G

must be assigned to the new index, which means that the new state of G will have one row more. In the example, splitting the second row of G resulted in the new index 5 which implies the addition of a new row to G. As a result, G with dimension four was updated to G^* with dimension five, while the dimension of y does not change.

In the same context of mixture models, Richardson and Green (1997) proposed a reversible jump approach to sample mixture representations when the number of components (parameters) is not only unknown but also inconstant. They considered six types of moves, some of which change the dimension of the parameter space. For the moves that keep this dimension unaltered, the acceptance probability reduces to the case of the M-H algorithm. In contrast, the moves that result in a change of dimension (e.g. split-merge) require a more elaborate procedure. To accomplish this, Richardson and Green (1997) implemented the reversible jump MCMC algorithm (RJMCMC) introduced by Green (1995). It was briefly discussed in Section 2.2.3.

The main feature of RJMCMC is the use of augmenting variables for matching dimensions and the definition of bijective transformations for ensuring reversibility of a move. That is, the existence of the inverse transformation is used for reversing the moves of the chain. Thus, a practical solution to the problem of the dimension change of G is to include a reversible jump step into the SMERED algorithm. The next chapter presents and develops this new idea. The new algorithm has been called SMERED$^+$.

4.3 Summary

Taken together, this chapter successfully resolves whether the GENUAD Markov chain converges to its invariant distribution. It has been shown that the algorithm produces an irreducible Markov chain which leads to the ergodicity of the chain. Irreducibility was the primary concern for GENUAD as the joint density of $(\mathcal{G}, X)$ does not satisfy the positivity condition, which can directly lead to irreducibility.

On the other hand, it was shown that the SMERED chain does not converge to its invariant distribution. This behaviour may be the result of the inadequate definition of a sampling distribution for G, which leads to a reducible chain. Alternatively, it may result as a consequence of inaccurately assuming a symmetric proposal distribution, which detrimentally alters the Metropolis ratio. Either way, considering the dimension change of G will enhance the performance of the split-merge operations in SMERED.

Chapter 5

New samplers

Chapter 3 presented two algorithms for addressing misidentification problems, where the uncertainty in the assignment of a unique and true identity of individuals is part of the model. The two algorithms are GENUAD by Wright et al. (2009) and SMERED by Steorts et al. (2016). The previous chapter presented the analysis of convergence for the algorithms, which showed convergence problems of the SMERED algorithm, due to the omission of the dimension change of the parameter G. This chapter presents the SMERED$^+$ algorithm. It is a modification of SMERED that considers that variation of the dimension along with a correct Metropolis ratio.

Also, another algorithm has been developed, called the DIU (Direct Identity Updater) algorithm. It is a Metropolized independent sampler (MIS, see Section 2.2.4), that directly updates the records in the sample. At each iteration, a single genotype is proposed for updating. Both SMERED$^+$ and DIU algorithms allow sampling from the distribution in Eq. (3.10), which refers to the true genotypes in the sample.

5.1 SMERED$^+$: Updating pairs of observations

The split-merge operations, on which the SMERED algorithm is based, increase or decrease the latent number of individuals in the observed sample, denoted by n. This is equivalent to saying that the dimension of G changes. The RJMCMC algorithm by Green (1995) explained in Section 2.2.3 allows moves between states with different dimension. Thus, a reversible jump step has been introduced into SMERED algorithm in order to account for the dimension change of G. The resulting algorithm, SMERED$^+$, is a trans-dimensional version of the SMERED algorithm by Steorts et al. (2016), and is outlined in Algorithm 3 below.

The algorithm comprises three main parts. First, the updater for (G, y), which implements the split-merge operations (steps 3-8). Second, the definition of r, which includes the transformations for the transdimensional jumps. Third, resampling G depending on the current value of y (step 11).

Algorithm 3 SMERED$^+$

Data: g^{obs}, N, p and γ
Initializers: G and y
Draw a pair of observed genotypes, say i and j for some $i \neq j$ in $\{1, \ldots, S\}$
if $y_i = y_j$ **then**
    Propose splitting that individual, shifting (G, y) to (G^*, y^*)
else
    Propose merging the individuals who i and j refer to, shifting (G, y) to (G^*, y^*)
end if
Calculate the corresponding ratio r (it will be defined later)
Set $(G^{\text{new}}, y^{\text{new}}) = (G^*, y^*)$ with probability $\min(1, r)$. Otherwise, $(G^{\text{new}}, y^{\text{new}}) = (G, y)$
Update G^{new} by using its full conditional density given y^{new}, shifting G^{new} to G^{newer}
return $G^{\text{newer}}, y^{\text{new}}$

5.1.1 Updater for G and y

According to Algorithm 2 for SMERED, G and y are jointly updated by using split-merge operations. The merge operation combines pairs of records by assigning them to the same individual in the population. The split operation separates two records, which are assigned to a common individual, into two different individuals. These operations are conducted taking into account the notion of compatibility (see Definition A.1.3) and co-reference between genotypes (see Definition A.1.4).

This section explains in detail the procedure for updating G and y jointly. Split-merge operations and the jumping distribution are the keys in this section. The split-merge operations in Steorts et al. (2016) start with the random choice of a pair of records. If the pair is associated with the same individual, a split will be proposed, otherwise they will be merged. The jumping distribution proposes new attributes for the split or merged individuals. The description in Algorithm 3 involves y but the use of X facilitates the computational design of SMERED$^+$.

5.1.1.1 Jumping distribution

The jumping distribution of SMERED$^+$ is categorical and depends on the new index (when merging) or the indices (when splitting) in y. Equivalently, it depends on the new row(s) of X. If g denotes a genotype and x a row in X, then the *jumping distribution* $J_m(g|x, g^{\text{obs}}, \gamma, p)$, for $m \in \mathcal{M}$, the set of possible moves, is determined as follows.

At each locus, find the number of unique alleles for each element of g^{obs} in which the proposed individual(s) was/were caught.

i. If this number exceeds 2, the proposed split is rejected as they cannot have come from the same individual (or, the proposed merge is rejected as they cannot be associated to the same individual).

ii. If this number is equal to 2, then the two alleles are assigned to the proposed genotype with probability equal to 1.0. This is because they are the same heterozygous genotype, which are the only known genotypes.

iii. If this number is equal to 1 then that allele is assigned to one of the alleles for the true genotype. For the second allele, the support is $1, 2, \ldots, m_l$ where m_l is the number of alleles at locus l. The corresponding probability has two factors, $\Pr(g_l|\gamma)$ and $\Pr(g_l^{\text{obs}}|g_l, p)$. The former is the corresponding value in the vector of allele frequencies γ. The later has the value $p_l/2$ if the resulting genotype is heterozygous or 1.0 if homozygous (Definition A.2.1).

This description characterizes the jumping distribution for the transdimensional approach as categorical.

5.1.1.2 Split-merge operations

As the data in Wright et al. (2009) are corrupted by the presence of allelic dropout, the number of suitable pairs of g^{obs} to be compared is limited. Only potential co-referent genotypes will be compared. There is no point in examining genotypes that never could be assigned to the same individual. An example of a pair with no chance to be co-referent is presented in Definition A.1.4.

Denote by $\mathcal{C}$ the set of all pairs of samples which could be co-referent, assuming that allelic dropout is the only source of genotyping error. Any in which the pair comes from the same known genotype is eliminated (i.e. pairs of genotypes that are heterozygous for all loci). At each update step, one element from $\mathcal{C}$ is chosen at random. If the pair is associated with the same individual, a split will be proposed, otherwise they will be merged. The operations are described as follows.

Splitting

If the two samples belong to a common individual indexed by i in the population, then the ith row of X, denoted by x_i, will be divided into two new rows, say x_1^+ and x_2^+, such that $x_i = x_1^+ + x_2^+$. This process will update X, whose number of rows has increased by one unit. This means that G has to be updated as well. For updating G, two new genotypes are obtained depending on x_1^+ and x_2^+ by using the jumping distribution $J_{\text{split}}(g|x, g^{\text{obs}}, \gamma, p)$ described above.

If the chosen samples match with the same latent individual, say some value i in $\{1, 2, \ldots, n\}$, then they are split following the steps below.

i. Find the collection $\mathcal{C}$ of observations associated with individual i, including the chosen.

ii. Retain the capture in one of the samples as being associated with individual i and propose allocating the capture in the other sample to a new index (not existing in the population), say i'.

iii. For each remaining observation in $\mathcal{C}$, randomly allocate the capture to either i or i'. If $|\mathcal{C}|= d$, there are 2^{d-2} ways of allocating these captures to the two individuals, existing and proposed.

iv. According to the partition in the previous step, the ith row of X has been updated by replacing it for two new rows denoted by x_1^+ and x_2^+. More precisely, they correspond to the rows i and i' in the new state of X.

v. Values for the corresponding rows in G must be proposed. Without loss of generality, denote them by g_1^+ and g_2^+, such that $g_1^+ \neq g_2^+$. They are assigned values by drawing independently from $J_{\text{split}}(g|x, g^{\text{obs}}, \gamma, p)$ described above, where x is the relevant row in X.

Merging

If the two chosen samples belong to different individuals, say indexed by i and j, then the ith and jth rows of X are merged into a single row, denoted by x^-, by summing them. That is, $x_i + x_j = x^-$. A row of X was deleted. While there are multiple ways of splitting, there is only one way of merging. The two genotypes associated with those two individuals are merged into a new genotype which is denoted by g^-. Then, the corresponding merged row of G needs to be updated. A value for g^- is found by drawing from $J_{\text{merge}}(g|x^-, g^{\text{obs}}, \gamma, p)$.

Example 5.1.1. Consider the observed genotypes g^{obs}, as below, with $S = 6$ samples and $L = 2$ loci.

$$g^{\text{obs}} = \begin{pmatrix} 1,1 & 1,1 \\ 1,2 & 2,2 \\ 1,2 & 1,3 \\ 1,1 & 1,1 \\ 3,3 & 1,1 \\ 2,2 & 3,3 \end{pmatrix}.$$

Assume that the latent information is given by

$$G = \begin{pmatrix} 1,2 & 2,3 \\ 1,2 & 1,3 \\ 2,3 & 1,2 \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Notice that the information contained in X can be expressed as $y = (2, 1, 2, 2, 3, 2)'$. Suppose that observed genotypes 1 and 4 are randomly chosen. As they belong to the same individual indexed by 2, then they are proposed for splitting. Then, the second row of X which is $(1, 0, 1, 1, 0, 1)'$ will be divided into two new rows.

Starting with the first step of the five steps described above, the collection of samples associated to the individual 2 is given by $\mathcal{C} = \{s_1, s_3, s_4, s_6\}$, where s_k denotes the sample k. The second step consists of assigning one of the chosen samples to a new index, say 4, while the other remain assigned to the current index which is 2. For example, suppose that sample 1 is chosen to be allocated to index 4 and sample 4 to

index 2. In the third step, the rest of samples in $\mathcal{C}$ are randomly allocated either to 2 or 4. This process partitions the set $\mathcal{C}$ into two sets. There are 2^2 ways to do this. All the possible partitions are listed below.

$$\begin{aligned}\mathcal{C} &= \{s_1\} \cup \{s_3, s_4, s_6\}\\ \mathcal{C} &= \{s_4\} \cup \{s_1, s_3, s_6\}\\ \mathcal{C} &= \{s_1, s_3\} \cup \{s_4, s_6\}\\ \mathcal{C} &= \{s_1, s_6\} \cup \{s_3, s_4\}\end{aligned}$$

For continuing with the fourth step, suppose that the resulting partition is the first one above. It means that the current row of X to be updated has been split into two new rows, $x_1^+ = (1,0,0,0,0,0)$ and $x_2^+ = (0,0,1,1,0,1)$. Lastly, g_1^+ and g_2^+ are independently drawn from $J_{\text{split}}(g|x_1^+, g^{\text{obs}}, \gamma, p)$ and $J_{\text{split}}(g|x_2^+, g^{\text{obs}}, \gamma, p)$, respectively. Table 5.1 shows the categories and their probabilities for sampling g_1^+. As stated above, the probabilities are computed as the product $\Pr(g^{\text{obs}}|g,p) \cdot \Pr(g|\gamma)$.

Table 5.1: Jumping distribution $J_{\text{split}}(g|x_1^+, g^{\text{obs}}, \gamma, p)$ for sampling g_1^+.

Category	$\Pr(g^{\text{obs}}\|g,p)$	$\Pr(g\|\gamma)$
$(1,1 \quad 1,1)$	1	$\gamma_{1,1}^{(1)} \cdot \gamma_{1,1}^{(2)}$
$(1,1 \quad 1,2)$	$1 \cdot p_2/2$	$\gamma_{1,1}^{(1)} \cdot \gamma_{1,2}^{(2)}$
$(1,1 \quad 1,3)$	$1 \cdot p_2/2$	$\gamma_{1,1}^{(1)} \cdot \gamma_{1,3}^{(2)}$
$(1,2 \quad 1,1)$	$p_1/2 \cdot 1$	$\gamma_{1,2}^{(1)} \cdot \gamma_{1,1}^{(2)}$
$(1,2 \quad 1,2)$	$p_1/2 \cdot p_2/2$	$\gamma_{1,2}^{(1)} \cdot \gamma_{1,2}^{(2)}$
$(1,2 \quad 1,3)$	$p_1/2 \cdot p_2/2$	$\gamma_{1,2}^{(1)} \cdot \gamma_{1,3}^{(2)}$
$(1,3 \quad 1,1)$	$p_1/2 \cdot 1$	$\gamma_{1,3}^{(1)} \cdot \gamma_{1,1}^{(2)}$
$(1,3 \quad 1,2)$	$p_1/2 \cdot p_2/2$	$\gamma_{1,3}^{(1)} \cdot \gamma_{1,2}^{(2)}$
$(1,3 \quad 1,3)$	$p_1/2 \cdot p_2/2$	$\gamma_{1,3}^{(1)} \cdot \gamma_{1,3}^{(2)}$

The distribution $J_{\text{split}}(g|x_2^+, g^{\text{obs}}, \gamma, p)$ for drawing g_2^+ has only one category, which is $(1,2, \quad 1,3)$ with probability 1.0. Then, after choosing a pair which resulted in a split, the new proposal may be set as,

$$G^* = \begin{pmatrix} 1,2 & 2,3 \\ 1,2 & 1,3 \\ 2,3 & 1,2 \\ 1,3 & 1,3 \end{pmatrix} \quad \text{and} \quad X^* = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

In this case, $y^* = (4,1,2,2,3,2)$. □

From Example 5.1.1, notice that the dimension of G and X changed, while the dimension of y remains unaltered. This change explains the use of X rather than y for constructing SMERED$^+$. However, the distribution of y will determine the Metropolis ratio.

5.1.2 The transdimensional approach

In analogy with the state space defined in Eq. (2.11), the elements in the state space of $f_{X,G}$ can be seen as pairs (X, G) indexed by X, where the dimension of G depends on X (equivalently y). More precisely, the number of rows n (distinct individuals observed in the sample) of G changes once X has been updated. If (X, G) denotes the current state of the Markov chain and a new move to (X^*, G^*) is proposed, then $n^* = n + 1$ when splitting, and $n^* = n - 1$ when merging. Clearly, the dimension of the parameter space is changing.

As explained in Section 2.2.3, RJMCMC introduces auxiliary random variables u generated from a convenient proposal distribution, and bijective functions used to match dimensions of (X, G) and (X^*, G^*). The dimension of (X, G) is defined as the number of rows n. If the current state is (X, G) and a split is proposed, then two auxiliary random variables u_1 and u_2 may be generated from a jumping distribution as defined in Section 5.1.1.1. For the reverse move (merge), a single auxiliary variable v is required. The approach here, which is not unique, is to use the identity transformation. That is,

$$\begin{aligned} u_1 &\leftrightarrow g_1^+ \\ u_2 &\leftrightarrow g_2^+ \\ v &\leftrightarrow g^- \end{aligned}$$

where g_1^+ and g_2^+ are any two rows in G^*, and g^- is a row in G.

This bijective transformation guarantees the dimension matching, for moving between (X, G) and (X^*, G^*) whose dimensions differ. Figure 5.1 illustrates a transformation h for splitting. Notice that G has three rows, while G^* has four rows. Then, two auxiliary random variables u_1 and u_2 are generated from the jumping distribution. Since the transformation h is defined such that the dimensions match, an auxiliary variable v is drawn in the image of the transformation. Now, the sum of dimensions is balanced, as Figure 5.1 shows.

In general, if the dimensions are defined as

$$\begin{aligned} \dim(X, G) &= n & \dim(u) &= r \\ \dim(X^*, G^*) &= n^* & \dim(u^*) &= r^* \end{aligned}$$

the transformation h maps (G, u) into (G^*, u^*) such that $n + r = n^* + r^*$. This equality is required to ensure the existence of the inverse of the transformation.

Metropolis-Hastings ratio

According to Eq. (2.14), the ratio for deciding whether the proposal is accepted or rejected is given by

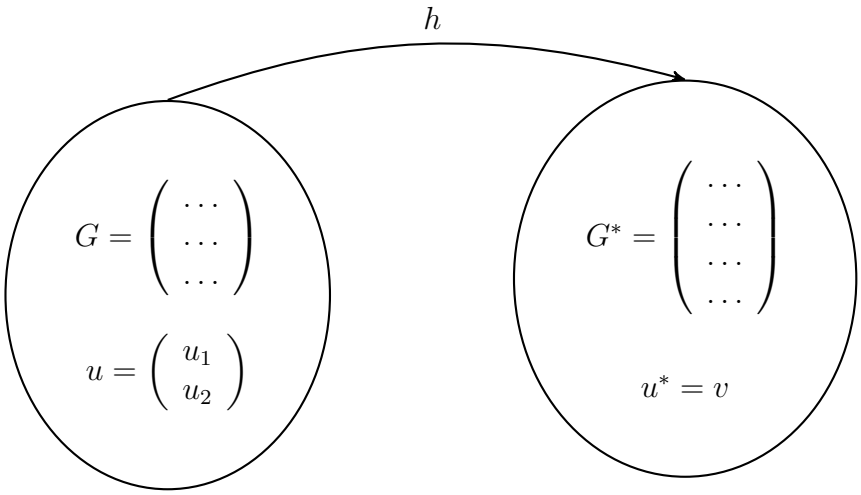


Figure 5.1: Representing a one-to-one transformation h for the case of a splitting.

$$r = \frac{f(g^{\text{obs}}|G^*, X^*, p) f(G^*|\gamma) f(y^*|N)}{f(g^{\text{obs}}|G, X, p) f(G|\gamma) f(y|N)} \cdot \frac{J_{X^* \to X}(u^*|g^{\text{obs}}, \gamma, p)}{J_{X \to X^*}(u|g^{\text{obs}}, \gamma, p)} \cdot \frac{j_{X^* \to X}}{j_{X \to X^*}} \tag{5.1}$$

It is known that the Jacobian factor in Eq. (2.14) results of applying the change of variables technique to the jumping distribution and the transformation. As the variables involved are discrete, there is no Jacobian term in this case.

The terms involved in the first ratio are defined in Wright et al. (2009). Indeed, $f(g^{\text{obs}}|G, X, p)$ is the sampling distribution and $f(G|\gamma)$ is the prior distribution for the parameter G. Most of the terms in the ratio $f(G^*|\gamma)/f(G|\gamma)$ will cancel. Only the terms that refer to the genotypes involved in the split-merge operation will remain. For example, if the current genotype is denoted g^{cur} is proposed for splitting, then

$$\frac{f(G^*|\gamma)}{f(G|\gamma)} = \frac{\Pr(g_1^+, g_2^+|\gamma)}{\Pr(g^{\text{cur}}|\gamma)} = \frac{\Pr(g_1^+|\gamma)\Pr(g_2^+|\gamma)}{\Pr(g^{\text{cur}}|\gamma)}.$$

In contrast, if two genotypes g_1^{cur} and g_2^{cur} are proposed for merging, then

$$\frac{f(G^*|\gamma)}{f(G|\gamma)} = \frac{\Pr(g^-|\gamma)}{\Pr(g_1^{\text{cur}}, g_2^{\text{cur}}|\gamma)} = \frac{\Pr(g^-|\gamma)}{\Pr(g_1^{\text{cur}}|\gamma)\Pr(g_2^{\text{cur}}|\gamma)}.$$

On the other hand, the prior probability of the model is given by the distribution of y, however it is the change in X which is treated as part of the new model description. From Eq. (3.9), for N fixed,

$$\frac{f(y^*|N)}{f(y|N)} = \begin{cases} N - n & \text{if splitting,} \\ (N - n + 1)^{-1} & \text{if merging.} \end{cases} \tag{5.2}$$

The second ratio in Eq. (5.1) corresponds to the jumping distribution. Three augmenting variables are required for balancing the dimensions of the parameter spaces.

They are denoted by u_1, u_2 for splitting and v for merging, and independently drawn from the jumping distribution J_m as explained before.

$$\begin{aligned} u_1 &\sim J_{\text{split}}(g|x_1^+, g^{\text{obs}}, \gamma, p), \\ u_2 &\sim J_{\text{split}}(g|x_2^+, g^{\text{obs}}, \gamma, p), \\ v &\sim J_{\text{merge}}(g|x^-, g^{\text{obs}}, \gamma, p), \end{aligned}$$

where x_1^+, x_2^+ and x^- denote rows of the indicator matrix X.

In the third ratio in Eq. (5.1), $j_{X\to X^*}$ represents the probability of a jump from X to X^*. As explained before, for the split case, if the number of captures associated with the row of X to be split is denoted by d, then there are 2^{d-2} ways of dividing them into two groups. Then, when splitting, $j_{X\to X^*} = 1/2^{d-2}$. When merging, there is only one way to do it. Then, the ratio is defined as follows.

$$\frac{j_{X^*\to X}}{j_{X\to X^*}} = \begin{cases} 2^{d-2} & \text{if splitting,} \\ 2^{-(d-2)} & \text{if merging.} \end{cases}$$

The set of ratios above define together the ratio in Eq. (5.1) for the specific model in Wright et al. (2009). Defining

$$\begin{aligned} r_1 &= \frac{f(g^{\text{obs}}|G^*, X^*, p)}{f(g^{\text{obs}}|X, G, p)} \cdot \frac{f(G^*|\gamma)}{f(G|\gamma)} \cdot \frac{f(X^*|N)}{f(X|N)} \\ r_2 &= \frac{J_{X^*\to X}(v|x^-, g^{\text{obs}}, \gamma, p)}{J_{X\to X^*}(u_1|x_1^+, g^{\text{obs}}, \gamma, p) \cdot J_{X\to X^*}(u_2|x_2^+, g^{\text{obs}}, \gamma, p)} \cdot \frac{j_{X^*\to X}}{j_{X\to X^*}} \end{aligned}$$

the Metropolis-Hastings ratio is given by

$$r = \begin{cases} r_1 \cdot r_2 & \text{if splitting,} \\ r_1 \cdot r_2^{-1} & \text{if merging.} \end{cases} \tag{5.3}$$

Notice how the augmenting variables and the transformations help to match the dimensions of the parameter spaces. To see this clearly, the ratio r is fully written below for the splitting case. Notice the balance in terms of the number of terms in the numerator and denominator.

$$\begin{aligned} r = \frac{f(g^{\text{obs}}|G^*, X^*, p) f(y^*|N)}{f(g^{\text{obs}}|G, X, p) f(y|N)} \cdot \frac{j_{X^*\to X}}{j_{X\to X^*}} \cdot \\ \cdot \frac{\Pr(g_1^+|\gamma)\Pr(g_2^+|\gamma) J_{X^*\to X}(v|x^-, g^{\text{obs}}, \gamma, p)}{\Pr(g^{\text{cur}}|\gamma) J_{X\to X^*}(u_1|x_1^+, g^{\text{obs}}, \gamma, p) \cdot J_{X\to X^*}(u_2|x_2^+, g^{\text{obs}}, \gamma, p)} \end{aligned}$$

The table below attempts to summarise the steps of SMERED$^+$ for updating jointly X and G. Assume that the current state of the chain is (X, G), and a new state (X^*, G^*) will be generated.

SMERED$^+$ algorithm: Jointly updating y and G
Starting at step $t-1$, the step t is obtained following the next steps:

1. Randomly choose a pair of observations, say i and j. If $y_i = y_j$ split them, otherwise merge them.
2. Use the jumping distribution $J_m(g|x, g^{\text{obs}}, \gamma, p)$ for generating augmenting variables (u_1, u_2, v) depending on the move m.
3. Map $u_1 \leftrightarrow g_1^+$, $u_2 \leftrightarrow g_2^+$ and $v \leftrightarrow g^-$, with g_1^+ and g_2^+ in G^* and g^- in G.
4. Calculate the ratio given by Eq. (5.3) and accept the proposal (y^*, G^*) with probability $\min(1, r)$.

5.1.3 Resampling G

Once y and G are jointly updated, the full conditional density defined in Section 3.2.2, $f(G|X, N, \gamma, p)$, is used for updating G separately. This step fixes the problem with step 9 of the original SMERED discussed in Section 4.2. Also, it shows that the difference between SMERED$^+$ and GENUAD for sampling from the posterior distribution in Eq. (3.10) is the joint updater for (y, G) in SMERED$^+$ and the updater for X in GENUAD.

5.1.4 Existence of the invariant distribution

By construction, a Markov chain generated from a RJMCMC algorithm has an invariant distribution. This statement is based on Section 2.2.3, where the acceptance probability was derived such that the proposal density satisfies the reversibility condition. This construction means that the Markov chain generated is reversible. Using Theorem 2.1.2, reversibility is a sufficient condition to ensure the existence of the stationary distribution. Failure to fulfil the reversibility condition could be considered a result of a poorly constructed proposal distribution. Example 4.2.1 for the original SMERED algorithm illustrated this situation.

There is no evidence whether the proposal (jumping) distribution in Section 5.1.1.1 for SMERED$^+$ is optimal. However, the choice was made such that the user knows where to jump safely. This refers to a sufficient condition of positivity of the proposal distribution J_m, that is, $J_m(g^{(t)}|g^{(t-1)}) > 0$. In other words, starting in $g^{(t-1)}$ the chain will jump to $g^{(t)}$ with positive probability. Moving to states for which this conditional probability is zero may result in a reducible chain. The critical point in the construction of that proposal distribution is the numeral (i.), because it would be the perfect scenario for proposing those illegal states. However, it was defined that under (i.), the proposal is rejected. Thus, the chain does not move to that proposal. Brooks et al. (2003), Hastie and Green (2012) and Farr et al. (2015) discuss techniques for improving proposals in RJMCMC.

5.2 DIU: Updating a single observation

This section introduces the DIU (Direct Identity Updater) algorithm, as an alternative approach for determining the real identities of the individuals in g^{obs}. It directly updates the true genotypes in the sample, denoted by g. From Definition 3.1.1, g is a deterministic function of G and y (equivalent to X). Indeed, the ith row of g is equal to the y_ith row of G. Denote by $\mathcal{U}$ the set of indices that give the collection of n unique genotypes in G (i.e. $\mathcal{U} \subseteq \{1, \ldots, S\}$ and $|\mathcal{U}| = n$). When $n < S$, $\mathcal{U}$ is not unique because different indices may provide the same set of unique genotypes.

As mentioned before, the M-H algorithm generates a Markov chain that moves through the state space using a proposal density which, in general, depends on the current state. However, it can be set as independent of the current state of the chain. This procedure is known as Metropolized independence sampler (MIS), already introduced in Section 2.2.4. The DIU algorithm is a MIS sampler.

Suppose that the Markov chain is currently in the state $g^{(t)}$, and the jth observation is proposed for updating, with $j = 1, \ldots, S$. Then, a proposal g_j^* is drawn from

$$J(g_j^*) \quad \propto \quad f(g_j^{\text{obs}} | g_j^*, p) \cdot f(g_j^* | \gamma) \tag{5.4}$$

where the first term in the right side is the likelihood function in Eq. (1.1) and the second term is the prior distribution for the true genotypes in the sample.

Making use of the independence between loci and updating one locus at a time, the proposal density is given by

$$\begin{aligned} J(g_j^*) &= J(g_{j1}^*, \ldots, g_{jL}^*) \\ &= \prod_{l=1}^{L} J(g_{jl}^*) \\ &= \prod_{l=1}^{L} \frac{f(g_{jl}^{\text{obs}} | g_{jl}^*, p) \cdot f(g_{jl}^* | \gamma)}{\sum_{g \in C} f(g_{jl}^{\text{obs}} | g, p) \cdot f(g | \gamma)} \\ &= \frac{f(g_j^{\text{obs}} | g_j^*, p) \cdot f(g_j^* | \gamma)}{\prod_{j=1}^{L} \sum_{g \in C} f(g_{jl}^{\text{obs}} | g, p) \cdot f(g | \gamma)}, \end{aligned}$$

where g_{jl}^* represents the latent pair of alleles for observation j at locus l, C is the set of genotypes compatible with g_{jl}^{obs}. This definition introduces the normalising constant (the denominator in the third equation). Although it is not required when defining the Metropolis ratio, it has important computational implications in the algorithm.

The Metropolis-Hastings ratio r is defined as

$$r \quad = \quad \frac{\pi(g^* | g^{\text{obs}}, N, \gamma, p)}{\pi(g^{(t)} | g^{\text{obs}}, N, \gamma, p)} \cdot \frac{J(g_k^{(t)})}{J(g_k^*)}$$

Using the posterior distribution π in Eq. (3.10) for g, since g is a deterministic function of G and X,

$$\pi(g|g^{\text{obs}}, N, \gamma, p) \propto f(g^{\text{obs}}|g, p) \cdot f(g|\gamma) \cdot f(y|N) \tag{5.5}$$

Then,

$$\begin{aligned} r &= \frac{\prod_{i=1}^{S} f(g_i^{\text{obs}}|g_i^*, p) \cdot \prod_{i \in \mathcal{U}^*} f(g_i^*|\gamma) f(y^*|N)}{\prod_{i=1}^{S} f(g_i^{\text{obs}}|g_i^{(t)}, p) \cdot \prod_{i \in \mathcal{U}^{(t)}} f(g_i^{(t)}|\gamma) f(y^{(t)}|N)} \cdot \frac{f(g_j^{\text{obs}}|g_j^{(t)}, p) \cdot f(g_j^{(t)}|\gamma)}{f(g_j^{\text{obs}}|g_j^*, p) \cdot f(g_j^*|\gamma)} \\ &= \frac{f(y^*|N)}{f(y^{(t)}|N)} \cdot \frac{f(g_j^{(t)}|\gamma)}{f(g_j^*|\gamma)} \cdot \frac{\prod_{i \in \mathcal{U}^*} f(g_i^*|\gamma)}{\prod_{i \in \mathcal{U}^{(t)}} f(g_i^{(t)}|\gamma)} \end{aligned} \tag{5.6}$$

where $\mathcal{U}^{(t)}$ is the set of indices providing the collection of $n^{(t)}$ unique genotypes. The cardinality of $\mathcal{U}^{(t)}$ is $n^{(t)}$. Similarly, $|\mathcal{U}^*| = n^*$. There are three cases for the proposal n^*:

1. $n^* = n^{(t)}$
2. $n^* = n^{(t)} + 1$
3. $n^* = n^{(t)} - 1$

If $n^* = n^{(t)}$, then the first ratio in Eq. (5.6) is equal to 1. There are three possible facts for the other two ratios:

i. $g_j^* = g_j^{(t)}$, which implies $r = 1$.

ii. $g_j^* \neq g_j^{(t)}$ and $\{g_i : i \in \mathcal{U}^{(t)}\} = \{g_i : i \in \mathcal{U}^*\}$. This situation arises when $g_j^* = g_k^{(t)}$ for some $k \in \{1, \ldots, S\}$, $k \neq j$. In this case, the third ratio in Eq. (5.6) is equal to 1 because the proposal resulted in a genotype already contained in $\{g_i : i \in \mathcal{U}^{(t)}\}$. So,

$$r = \frac{f(g_j^{(t)}|\gamma)}{f(g_j^*|\gamma)}$$

iii. $g_j^* \neq g_j^{(t)}$ and $\{g_i : i \in \mathcal{U}^{(t)}\} \neq \{g_i : i \in \mathcal{U}^*\}$. This is the case of $g_j^* \neq g_k^{(t)}$ for all $k \in \{1, \ldots, S\}$, $k \neq j$. That is, the proposal does not exist in the current set of unique genotypes. Then, $r = 1$ because

$$\frac{\prod_{i \in \mathcal{U}^*} f(g_i^*|\gamma)}{\prod_{i \in \mathcal{U}^{(t)}} f(g_i^{(t)}|\gamma)} = \frac{f(g_j^*|\gamma)}{f(g_j^{(t)}|\gamma)}$$

Now, if $n^* = n^{(t)} + 1$, the third ratio in Eq. (5.6) is equal to $f(g_j^*|\gamma)$ and r reduces to

$$r = \frac{f(y^*|N)}{f(y^{(t)}|N)} \cdot f(g_j^{(t)}|\gamma)$$

If $n^* = n^{(t)} - 1$, the third ratio in Eq. (5.6) is equal to $1/f(g_j^{(t)}|\gamma)$ and r reduces to

$$r = \frac{f(y^*|N)}{f(y^{(t)}|N)} \cdot \frac{1}{f(g_j^*|\gamma)}$$

Therefore, the Metropolis ratio for the DIU algorithm is defined by

$$r = \begin{cases} \dfrac{f(g_j^{(t)}|\gamma)}{f(g_j^*|\gamma)} & \text{if } n \text{ does not change and condition (ii.) holds,} \\ f(g_j^{(t)}|\gamma) \cdot (N - n^{(t)}) & \text{if } n \text{ increases,} \\ \dfrac{1}{f(g_j^*|\gamma)} \cdot \dfrac{1}{N - n^{(t)} + 1} & \text{if } n \text{ decreases,} \\ 1 & \text{otherwise.} \end{cases} \tag{5.7}$$

The proposal g_j^* is accepted with probability equal to $\min(1, r)$, where r is defined by Eq. (5.7). If the proposal is accepted, $g_j^{\text{new}} = g_j^*$. Otherwise, $g_j^{\text{new}} = g_j^{(t)}$. The matrix resulting from this process is g^{new}, which is different from $g^{(t)}$ up to a single row. The unique genotypes in the new state g^{new} are denoted by G^{new}, which is updated using its full conditional density. This density is conditional on the current value of y. The new state, denoted by $G^{(t+1)}$, gives a g matrix denoted by $g^{(t+1)}$. Algorithm 4 outlines the steps followed by DIU.

The DIU algorithm will produce a Markov chain whose states are matrices g containing the true genotypes in g^{obs}. Step 7 in Algorithm 4 has been included to align all the algorithms. This step is analogous to step 11 in Algorithm 3 for SMERED$^+$, and step 4 in Algorithm 1 for GENUAD. This step allows to two consecutive g matrices to be different. However, their number of unique genotypes may be either the same or differ by one unit.

Algorithm 4 DIU (Direct Identity Updater)

1: **Data:** g^{obs}, N, p and γ
2: **Initializers:** g
3: Choose at random a row of g, say j
4: Draw g_j^* from the proposal density defined by Eq. (5.4)
5: Calculate r as defined in Eq. (5.7).
6: Set $g^{\text{new}} = g^*$ with probability $\min(1, r)$. Otherwise, $g^{\text{new}} = g$
7: Update G^{new} by using its full conditional density given g^{new}, shifting G^{new} to G^{newer}
8: **return** g^{newer}

The following example shows how the DIU algorithm updates a single row of g.

Example 5.2.1. Consider g^{obs} as in Example 5.1.1. The matrix g with the true genotypes in g^{obs} is deterministically found from G and X. Then,

$$g = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 1,2 & 1,3 \\ 1,2 & 1,3 \\ 2,3 & 1,2 \\ 2,3 & 1,3 \end{pmatrix} \quad \text{and} \quad y = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \\ 3 \\ 4 \end{pmatrix}.$$

Notice that currently $n = 4$. Suppose that row 6 has been randomly chosen for which $g_6^{\text{obs}} = (2,2 \quad 3,3)$. Then, a proposal g_6^* is drawn from the density in Eq. (5.4). Table 5.2 shows the categorical distribution. The probabilities of the categories are given by the product $\Pr(g^{\text{obs}}|g,p) \cdot \Pr(g|\gamma)$.

Table 5.2: Proposal distribution of DIU for sampling the 6th genotype in g.

	Category	$\Pr(g^{\text{obs}}\|g,p)$	$\Pr(g\|\gamma)$
Locus 1	$(1,2)$	$p_1/2$	$\gamma_{1,2}^{(1)}$
	$(2,2)$	1	$\gamma_{2,2}^{(1)}$
	$(2,3)$	$p_1/2$	$\gamma_{2,3}^{(1)}$
Locus 2	$(1,3)$	$p_2/2$	$\gamma_{1,3}^{(2)}$
	$(2,3)$	$p_2/2$	$\gamma_{2,3}^{(2)}$
	$(3,3)$	1	$\gamma_{3,3}^{(2)}$

From the independence among loci, there are $3^2 = 9$ possible combinations of alleles pairs from which one is randomly sampled. Each of the cases for n^* is illustrated as follows.

- For (i.) in case 1, the proposal g_6^* is equal to the current genotype. That is, $g_6^* = (2,3 \quad 1,3)$ is sampled with probability $p_1 \cdot p_2 \cdot \gamma_{2,3}^{(1)} \cdot \gamma_{1,3}^{(2)}/4$. In this case, $r = 1$.

- For (ii.) in case 1, the proposal is different to the current genotype, but equal to one element in the set of unique genotypes (i.e. n does not change). For example, $g_6^* = (1,2 \quad 1,3)$ is sampled with probability $p_1 \cdot p_2 \cdot \gamma_{1,2}^{(1)} \cdot \gamma_{1,3}^{(2)}/4$. From Eq. (5.7),

$$r = \frac{f(g_6|\gamma)}{f(g_6^*|\gamma)} = \frac{\gamma_{2,3}^{(1)} \cdot \gamma_{1,3}^{(2)}}{\gamma_{1,2}^{(1)} \cdot \gamma_{1,3}^{(2)}} = \frac{\gamma_{2,3}^{(1)}}{\gamma_{1,2}^{(1)}}$$

- For (iii.) in case 1, the proposal is different to all current unique genotypes, but it does not change the current value of n. For example, $g_6^* = (2,3 \quad 3,3)$ is sampled with probability $p_1 \cdot \gamma_{2,3}^{(1)} \cdot \gamma_{3,3}^{(2)}/2$. In this case, $r = 1$.

- For case 3, the current value of n decreases. For example, $g_6^* = (1,2 \quad 2,3)$ may be sampled with probability $p_1 \cdot p_2 \cdot \gamma_{1,2}^{(1)} \cdot \gamma_{2,3}^{(2)}/4$. From Eq. (5.7) with $n = 4$,

$$r = \frac{1}{f(g_6^*|\gamma)} \cdot \frac{1}{N-n+1} = \frac{1}{\gamma_{1,2}^{(1)} \cdot \gamma_{2,3}^{(2)}} \cdot \frac{1}{N-n+1}$$

The case 2 cannot be illustrated using row 6 in g. Suppose that row 2 is chosen for updating. Table 5.3 shows the categorical distribution in this case. At locus 1, the allele pair 1,2 is chosen with probability 1.0, as a heterozygous was observed. A proposal may be $g_2^* = (1,2 \quad 1,2)$ with probability $p_2 \cdot \gamma_{1,2}^{(2)}/2$. This proposal increases $n = 4$ to $n^* = 5$. From Eq. (5.7), $r = f(g_2|\gamma) \cdot (N - n)$ where $n = 4$.

Table 5.3: Proposal distribution of DIU for sampling the 2nd genotype in g.

	Category	$\Pr(g^{\text{obs}}\|g,p)$	$\Pr(g\|\gamma)$
Locus 2	$(1,2)$	$p_2/2$	$\gamma_{1,2}^{(2)}$
	$(2,2)$	$p_2/2$	$\gamma_{2,2}^{(2)}$
	$(2,3)$	1	$\gamma_{2,3}^{(2)}$

For all cases above, r will determine whether the proposal is accepted or rejected. If it is accepted, $g^{\text{new}} = g^*$. Otherwise, $g^{\text{new}} = g$.

For example, if the proposal is accepted in case 3 above, g^{new} and the corresponding G^{new} are given by

$$g^{\text{new}} = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 1,2 & 1,3 \\ 1,2 & 1,3 \\ 2,3 & 1,2 \\ 1,2 & 2,3 \end{pmatrix} \quad \text{and} \quad G^{\text{new}} = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 2,3 & 1,2 \end{pmatrix}.$$

Note that the new current value of y is $y^{\text{new}} = (1,2,1,1,3,2)'$. Also, g^{new} and the current g differ only in row 6. Using step 7 in Algorithm 4, G^{new} is updated given y^{new}. Suppose that G^{newer} is given by

$$G^{\text{newer}} = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 3,3 & 1,1 \end{pmatrix}.$$

The pair G^{newer} and y^{new} produce the new updated value of g as follows.

$$g^{\text{newer}} = \begin{pmatrix} 1,2 & 1,3 \\ 1,2 & 2,3 \\ 1,2 & 1,3 \\ 1,2 & 1,3 \\ 3,3 & 1,1 \\ 1,2 & 2,3 \end{pmatrix}$$

The matrices g and g^{newer} differ in two rows but the difference between the number of unique genotypes is 1. It passed from $n = 4$ to $n^{\text{newer}} = 3$. □

Properties of DIU

The DIU algorithm is a Metropolis independent sampler, introduced in Section 2.2.4. In contrast to the other Metropolis algorithms, for the case of finite sample spaces, the actual transition matrix of the Markov chain generated by DIU can be found. This allows the eigenvalues and eigenvectors to be determined for studying the convergence to the invariant distribution. The most informative in this sense is the second largest eigenvalue, and its importance lies in the fact that it provides information on the mixing rate of the chain (see Liu (1996), Liu (2008)). Liu (1996) shows results for this particular case of Metropolis sampling, procedures for finding eigenvalues and the corresponding eigenvectors, and an upper bound for the L^1 distance between the target and the simulated distribution.

Eigenvalue analysis for DIU

Let $\mathcal{X}_g$ denote the space state of g of a Markov chain generated by DIU. Suppose that $\mathcal{X}_g$ is finite with q distinct states. The states in $\mathcal{X}_g$ are labelled according to the values of their importance ratios, defined in Eq. (2.15). That is,

$$w_1 \geq w_2 \geq \ldots \geq w_q$$

where w_i is the importance ratio of a state $g \in \mathcal{X}_g$ with the ith largest importance ratio. Liu (2008) uses the notation $w_i = w(i)$. For DIU, $w = \pi(g|g^{\text{obs}}, N, \gamma, p)/J(g)$ where $J(\cdot)$ refers to the proposal density defined in Eq. (5.4).

The transition matrix is explicitly expressed as

$$K = \begin{pmatrix} J_1 + \lambda_1 & \pi_2/w_1 & \pi_3/w_1 & \ldots & \pi_{q-1}/w_1 & \pi_q/w_1 \\ J_1 & J_2 + \lambda_2 & \pi_3/w_2 & \ldots & \pi_{q-1}/w_2 & \pi_q/w_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ J_1 & J_2 & J_3 & \ldots & J_{q-1} + \lambda_{q-1} & \pi_q/w_{q-1} \\ J_1 & J_2 & J_3 & \ldots & J_{q-1} & J_q \end{pmatrix}$$

where $\lambda_k = \sum_{i=k}^{q}(J_i - \pi_i/w_k)$ is the probability of a rejection in the next step if the current state is k. It is shown in Liu (1996) that the eigenvalues for K are $1 > \lambda_1 \geq \lambda_2 \geq \lambda_{q-1}$.

The second largest eigenvalue λ_1 is expressed as $1 - 1/w_1$, and asymptotically controls the mixing rate of the chain, when $\mathcal{X}_g$ is finite and the number of iterations is large. An upper bound for the *total variation distance* between J and π is provided in Liu (1996). The lemma is omitted here, but the bound has the form λ_1^{2T}/π_x, where x is any starting state and T is number of iteration. Additional bounds are given considering other eigenvalues, which luckily have explicit expressions. The use of the coupling method also provides bounds for this distance (see Liu (1996), Liu (2008)). The upper bound is given by $\left(1 - \frac{1}{w_1}\right)^T$.

However, Metropolis independent sampling is an exceptional case, because in general, determining the structure of the transition matrix is virtually impossible. Therefore, only finding estimates for upper and lower bounds of the second largest value is achievable. For example, a bottleneck is a subset of the state space that makes portions of it difficult to reach from some starting locations, which limits the speed of convergence. Consequently, bottlenecks might control the mixing time of the chain. Levin et al. (2009) provides lower bounds for the mixing time for a Markov chain with bottlenecks.

The Perron-Frobenius theorem also helps to measure mixing time. It states that the convergence to the invariant distribution of an ergodic Markov chain in a finite space state is geometric, with relative speed equal to the second largest eigenvalue. More details can be found in Levin et al. (2009), Behrends (2000), Brémaud (1999).

Regarding the existence of the invariant distribution, the argument is based on the fact that DIU is a particular case of the M-H algorithm. Similar to the conclusion for SMERED$^+$, the reversibility implies the existence of the invariant distribution. Irreducibility of the chain results from the positivity of the density in Eq. (5.4), as stated by Robert and Casella (2004). For the case of DIU, the conditional density $J(g)$ defined in Eq. (5.4) should satisfy positivity. This condition allows to conclude that the chain is irreducible. Equivalently, if both conditional densities $f(g^{\text{obs}}|g,p)$ and $f(g|\gamma)$ satisfy the positivity condition then irreducibility is easily concluded. Indeed, they are positive by definition as given in Wright et al. (2009), equations 1-4. In other words, from equations 1-3, given g and p, $f(g^{\text{obs}}|g,p) > 0$ for all g^{obs}. And from equation 4, knowing γ, $f(g|\gamma) > 0$ for all g.

5.3 Summary

This chapter presented two new algorithms for sampling from the posterior density in Eq. (3.10), called SMERED$^+$ and DIU. This density refers to the observable part of the misidentification problem in Wright et al. (2009). SMERED$^+$ is an RJMCMC algorithm for taking into account the dimension change of G. The procedure randomly chose pairs of observations for splitting or merging, depending on the current status. These split and merge operations update not only the pair but also all other observations linked to them. Instead, DIU randomly chose a single row of g. It is a Metropolized independent sampler, that is, the proposal is an independent transition function from the current state. In this particular case, the analysis of all the eigenvalues of the transition matrix is feasible for a finite state space. For both SMERED$^+$ and DIU, the existence of the invariant distribution is ensured by construction because the Metropolis ratio is defined such that reversibility holds.

Chapter 6

Applications to Genetic Data

Previous chapters presented three different algorithms (GENUAD, SMERED$^+$, and DIU) for simulating the posterior distribution in Eq. (3.10). This chapter implements these algorithms using three examples which rely on the badger records data considered by Wright et al. (2009). In the first example, the number of observations and loci have been markedly reduced for illustrative purposes. The second and third examples provide a more realistic situation, as they consider the entire data set. The difference between the two datasets is the number of replicates during the PCR that was carried out for the DNA amplification. The underlying purpose of the replicates is to "clean" the biological data as much as possible; the more replicates that are carried out, the cleaner the data is. The second example considers data which contains two replicates, and the third examines the case with two or more replicates.

Although the toy example presented below may be too restrictive, it is necessary to explain the performance of the algorithms clearly and to provide some intuition about them. Albeit, this intuition might not apply to high dimensional data. For instance, the state space associated with the true genotypes in the sample of the toy example is minimal. These settings imply an availability of the exact posterior target distribution for contrasting with the simulated distributions by the three algorithms. This comparison is not possible with the larger dataset because the state space is extensive. Thus, the analysis of the full data set needs to be addressed differently. In the other two examples, different diagnostic tests in the `CODA` library of the `R` software were applied to assess convergence. The aim was to detect failures in the convergence to a stationary state rather than to "prove" convergence. The `R` and `C` code for fitting these models was written by Prof. Richard Barker, Dr. Chris Stevens and Nick Gelling of the University of Otago.

Therefore, Section 6.1 introduces the results of applying the algorithms to the toy example, Section 6.2 considers the two replicates data, and Section 6.3 presents the results when performing two or more replicates.

6.1 Toy example

In this section, Example 3.2.1 is used to compare the GENUAD, SMERED$^+$, and DIU algorithms. The main goal is to sample from the posterior distribution in Eq. (3.10). Because it is a small-scale example, the state space is small, which means that the posterior distribution simulated by the algorithms can be compared with the exact posterior distribution.

Consider $S = 3$ samples and $L = 2$ loci. The observed genotypes g^{obs} are given by

$$g^{\text{obs}} = \begin{pmatrix} 1,1 & 1,1 \\ 1,2 & 2,2 \\ 1,2 & 1,3 \end{pmatrix}.$$

The settings for this example are as follows.

- The population size, $N = 5$.
- The number of alleles at each loci, $m = (2, 3)$.
- The number of genotypes at each loci, $\eta = (3, 6)$.
- The dropout probabilities at each loci, $p = (0.25, 0.35)$.
- The allele frequencies at locus 1, $\gamma^{(1)} = (1/3, 1/3, 1/3)$.
- The allele frequencies at locus 2, $\gamma^{(2)} = (1/6, 1/6, 1/6, 1/6, 1/6, 1/6)$.

Notice that the allele frequencies at each locus have been established as equally probable.

Set $\mathcal{G}^{(0)}$ and $X^{(0)}$ as initial states such that $(\mathcal{G}^{(0)}, X^{(0)}) \in \text{supp}(f_{\mathcal{G},X})$ as below.

$$\mathcal{G}^{(0)} = \begin{pmatrix} 1,2 & 1,2 \\ 1,2 & 1,3 \\ 1,2 & 2,3 \\ 2,2 & 2,3 \\ 1,1 & 1,2 \end{pmatrix} \quad \text{and} \quad X^{(0)} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The true genotypes in g^{obs} associated with $(\mathcal{G}^{(0)}, X^{(0)})$ and the vector of indices are given by

$$g^{(0)} = \begin{pmatrix} 1,2 & 1,2 \\ 1,2 & 1,2 \\ 1,2 & 1,3 \end{pmatrix} \quad \text{and} \quad y^{(0)} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}.$$

Recall that, in general, the ith row of g is equal to y_ith row of $\mathcal{G}$, that is, $g_i = \mathcal{G}_{y_i}$. The matrix $g^{(0)}$ is one of the 18 possible matrices in the state space of g, $\mathcal{X}_g$. This number comes from counting the number of possibilities for the observed homozygous genotypes in g^{obs} ($2 \cdot 3^2 = 18$). Table 6.1 shows the elements of $\mathcal{X}_g$. Each row represents

a matrix g when filling by rows. For example, for $i = 10$, the row represents the matrix below. For $i \in \{4, 6, 12, 18\}$, the corresponding g's have $n = 2$, for the rest $n = 3$.

$$g = \begin{pmatrix} 1,2 & 1,2 \\ 1,2 & 2,2 \\ 1,2 & 1,3 \end{pmatrix}.$$

Table 6.1: Elements in the state space of g, $\mathcal{X}_g$.

i	Sample 1				Sample 2				Sample 3			
1	1	1	1	1	1	2	1	2	1	2	1	3
2	1	2	1	1	1	2	1	2	1	2	1	3
3	1	1	1	2	1	2	1	2	1	2	1	3
4	1	2	1	2	1	2	1	2	1	2	1	3
5	1	1	1	3	1	2	1	2	1	2	1	3
6	1	2	1	3	1	2	1	2	1	2	1	3
7	1	1	1	1	1	2	2	2	1	2	1	3
8	1	2	1	1	1	2	2	2	1	2	1	3
9	1	1	1	2	1	2	2	2	1	2	1	3
10	1	2	1	2	1	2	2	2	1	2	1	3
11	1	1	1	3	1	2	2	2	1	2	1	3
12	1	2	1	3	1	2	2	2	1	2	1	3
13	1	1	1	1	1	2	2	3	1	2	1	3
14	1	2	1	1	1	2	2	3	1	2	1	3
15	1	1	1	2	1	2	2	3	1	2	1	3
16	1	2	1	2	1	2	2	3	1	2	1	3
17	1	1	1	3	1	2	2	3	1	2	1	3
18	1	2	1	3	1	2	2	3	1	2	1	3

The following is a review of how each algorithm updates the initial state $(\mathcal{G}^{(0)}, X^{(0)})$.

GENUAD updates $\mathcal{G}^{(0)}$ and $X^{(0)}$ by a Gibbs sampler. The former is updated by rows (individuals in the population) and the later by columns (individuals in the sample). The conditional densities have been explained in Section 3.2.2. The Markov chain then generates a sequence of values for $(\mathcal{G}, X)$ which leads to g.

For example, suppose that $X^{(0)}$ is updated given $\mathcal{G}^{(0)}$, resulting in the new state X^* as below. Conditioned on X^*, $\mathcal{G}^*$ is obtained. Thus, the pair $(\mathcal{G}^*, X^*)$ determines the state g^* for the true genotypes in g^{obs}.

$$X^* = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathcal{G}^* = \begin{pmatrix} 1,2 & 2,3 \\ 1,2 & 1,3 \\ 1,1 & 1,2 \\ 2,2 & 2,2 \\ 1,1 & 1,3 \end{pmatrix} \quad \Rightarrow \quad g^* = \begin{pmatrix} 1,1 & 1,3 \\ 1,2 & 2,3 \\ 1,2 & 1,3 \end{pmatrix}$$

SMERED$^+$ starts with the pair $(G^{(0)}, y^{(0)})$, where $G^{(0)}$ contains the unique genotypes in the sample and defined as

$$G^{(0)} = \begin{pmatrix} 1,2 & 1,2 \\ 1,2 & 1,3 \end{pmatrix}.$$

These initial states are jointly updated by a RJMCMC move. As explained in Section 5.1.1, the updater starts by taking a random pair of samples in g^{obs} and updating the index (or indices) associated with that choice. All of the genotypes associated with those indices are also updated in the same step. For example, suppose the initial state described above. If samples 1 and 2 are chosen for updating, then they are proposed for splitting, as they are currently assigned to the first individual in G. Only the genotypes associated with this individual are updated. Suppose the new values are,

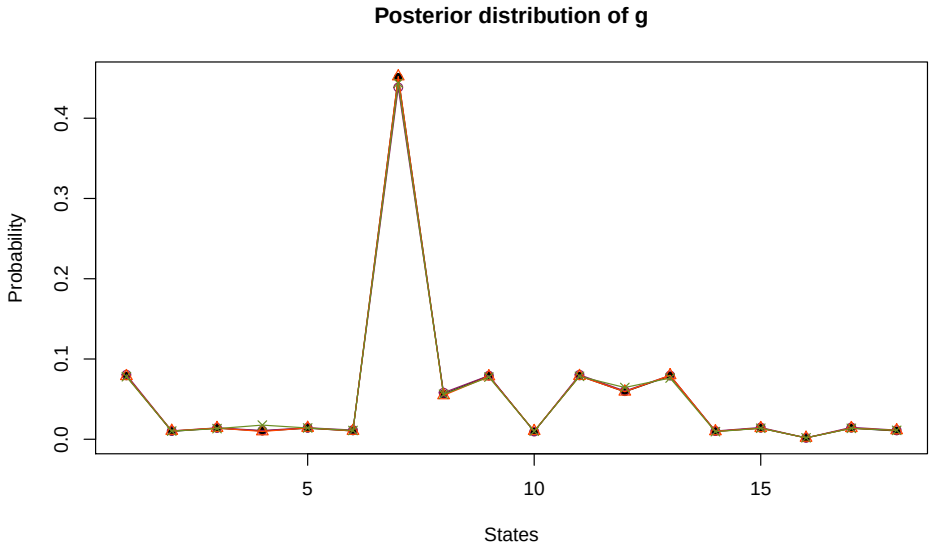
$$y^* = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}, \qquad G^* = \begin{pmatrix} 1,1 & 1,3 \\ 1,2 & 1,3 \\ 1,2 & 2,3 \end{pmatrix} \quad \Rightarrow \quad g^* = \begin{pmatrix} 1,1 & 1,3 \\ 1,2 & 2,3 \\ 1,2 & 1,3 \end{pmatrix}.$$

DIU starts with the state $g^{(0)}$. At each iteration, a single row of g is randomly chosen for updating. For example, the state g^* as above can be obtained if either row 1 or 2 are chosen. Then, the proposal distribution defined in Eq. (5.4) along with step 7 may lead to the new matrix g^* as above.

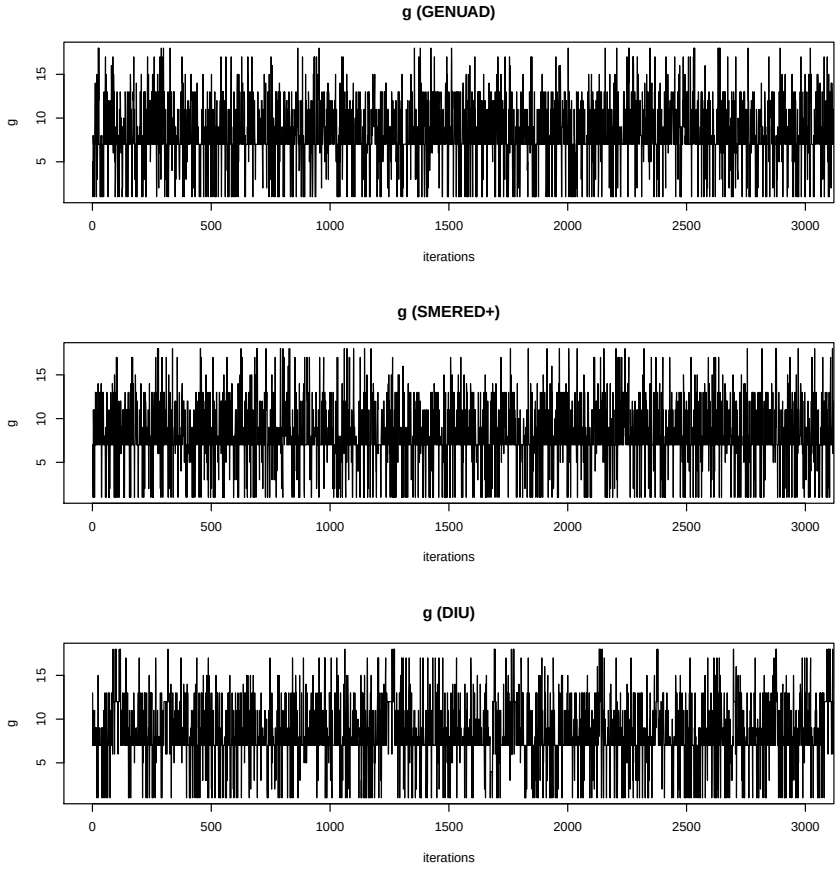
The use of the matrix g^* as a destination state is a deliberate example to show how different updaters reach the same state in a single step. Although in practice this is unlikely, it illustrates that GENUAD algorithm updates separately $\mathcal{G}$ and X, SMERED$^+$ jointly updates G and y, and DIU directly updates g.

Now, because $\mathcal{X}_g$ is small, the probability distribution of g can be found by using the expression in Eq. (3.10). The exact values of this posterior distribution in addition to the probabilities of visiting the states in $\mathcal{X}_g$ are shown in Figure 6.1(a). The stationary distribution associated with the Markov chains generated by the algorithms are similar to the exact distribution of g.

The inspection of the trace plots in Figure 6.1(b) for assessing the convergence of the Markov chains shows that they traverse $\mathcal{X}_g$ rapidly because they have the ideal shape (similar to an "accordion" shape). That is, the chain does not visit the same state for extended periods, and the steps do not tend to follow a particular direction. This feature is a signal of good mixing in the space $\mathcal{X}_g$. However, there is one difficulty with this interpretation that may conceal the actual performance of the chains, which is the lack of a distance notion in $\mathcal{X}_g$. The 18 labels in the vertical axis are just that, labels that were assigned arbitrarily to all states in $\mathcal{X}_g$. Thus, there is not a notion of order, which implies that another assignment of labels could give a completely different trace plot, perhaps unfavourable for the algorithms. However, these trace plots can report if the chains get stuck at states of $\mathcal{X}_g$ which is not the case.



(a) Comparing the exact distribution and the simulated by GENUAD, SMERED$^+$ and DIU algorithms.



(b) Trace plots for the 18 states of g. Only the first 3000 iterations are shown.

Figure 6.1: Exploring the state space $\mathcal{X}_g$.

As an exercise in preparation for the second example, consider the number of unique rows of g, denoted by n. This quantity n can be used as a summary of g. Table 6.2 shows the posterior distribution of n, which shows similar probabilities for the three chains. Also, Figure 6.2 shows the behaviour of the chains for exploring the values of n through the first 3000 iterations. GENUAD and SMERED$^+$ move relatively easily between $n = 2$ and $n = 3$. However, DIU often gets stuck at $n = 3$, which is the value of n with the largest frequency. Although this example is small, notice that the summary n is a numerical variable, and so the concept of closeness between states makes sense. Taken together, in this example, GENUAD and SMERED$^+$ seem to explore $\mathcal{X}_n$, the state space of n, relatively well while DIU explored the state space poorly.

Table 6.2: Posterior distribution of n

n	GENUAD	SMERED$^+$	DIU
2	0.09395	0.09039	0.10417
3	0.90605	0.90961	0.89583

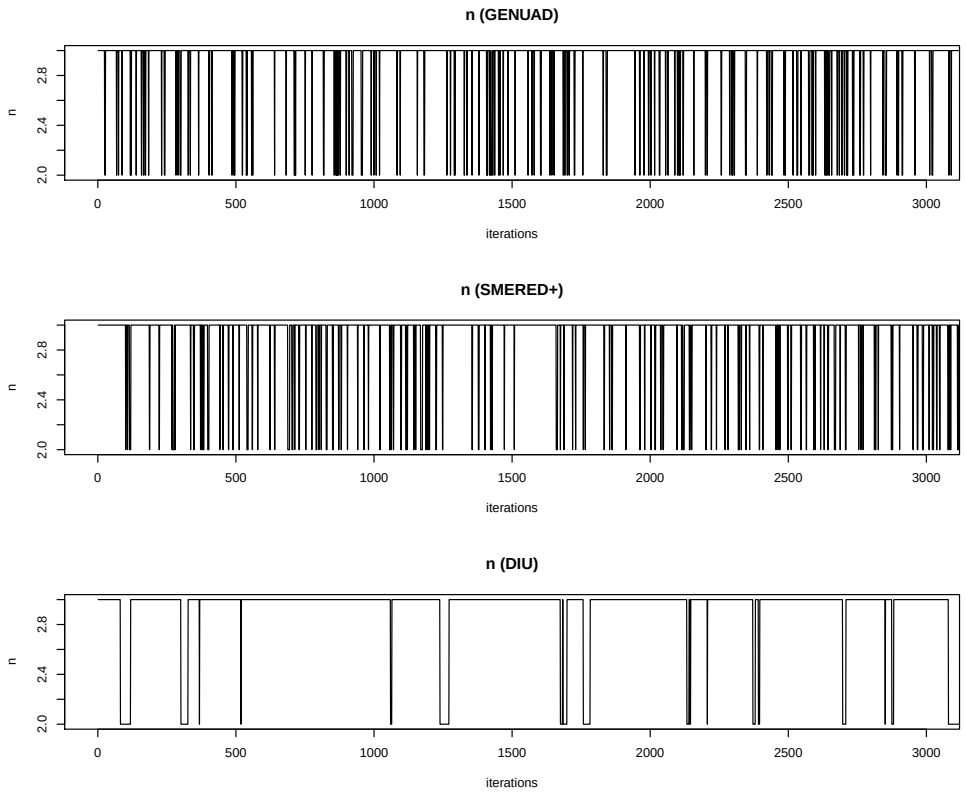


Figure 6.2: Trace plots for n. Only the first 3000 iterations are shown.

The toy example serves as a prelude to the full dataset considered by Wright et al. (2009) because it illustrates, at a small scale, the relevant state spaces involved in the simulation, that is, $\mathcal{X}_g$ and $\mathcal{X}_n$. The next section presents the results obtained for the entire dataset.

6.2 Two PCR replicates

Here, the GENUAD, SMERED$^+$, and DIU algorithms have been implemented using the dataset considered by Wright et al. (2009), which consider two replicate PCR amplifications ($R = 2$). After having addressed the toy example in the previous section, it is slightly easier to explain how the algorithms work for the full dataset.

The matrix g^{obs} has $S = 47$ observed genotypes at $L = 7$ loci, for which the number of alleles is given by $m = (6, 4, 5, 4, 4, 4, 3)$. The number of possible genotypes at each locus is determined by using the formula $\eta_l = m_l(m_l + 1)/2$ for $l = 1, \ldots, L$, which define the vector $\eta = (21, 10, 15, 10, 10, 10, 6)$. The fixed values are $N = 40$ and $p = (0.5, 0.8, 0.3, 0.2, 0.6, 0.4, 0.7)$. The allele frequencies in γ have been considered as equally probable at each locus (similarly to the toy example). Although N, p and γ are estimated in Wright et al. (2009), they are fixed here as the focus is the mechanism for updating G and X. These settings should not alter the relative performance of the algorithms.

As seen in the toy example, there are two state spaces of interest. One is the state space of the matrices g (the true genotypes in the sample), denoted by $\mathcal{X}_g$, and the other is the state space of n (the number of unique individuals in the sample), denoted by $\mathcal{X}_n$. In this example, the size of $\mathcal{X}_g$ is extremely large (3.22681×10^{119} elements) and cannot feasibly be managed as in the toy example. Thus, the adoption of n as a summary of g is a practical and convenient solution, since n is a discrete variable which might be equal to 1 (all the observed genotypes belong to the same individual) and up to S (all the observed genotypes belong to different individuals). The state space of n, $\mathcal{X}_n$, would thus be considerably smaller.

For each algorithm, two Markov chains for g were generated. For 200 000 iterations of the algorithms, the chains have a thinning interval of 10, which means that they are 20 000 long. Thinning the chain has the sole purpose of reducing storage. For SMERED$^+$, the first 20 000 iterations (first 200 storaged values) were discarded. The initial states g have $n = 19$ and $n = 34$ unique individuals.

Exploring the state space $\mathcal{X}_n$

The following is a comparative analysis of the chains generated by the GENUAD, SMERED$^+$ and DIU algorithms. The aim is to understand their differences concerning convergence properties. The trace plots and the diagnostics of convergence explained in Section 2.3 are used.

Two chains starting at different values of n were simulated. Figure 6.3 shows the overlapped trace plots starting at $n = 19$ (cyan) and $n = 34$ (orange). The trace plot is the first diagnostic by default to detect convergence issues of the Markov chains. Ideally, a chain has a satisfactory trace plot if, when framing a portion of iterations, the plot behaves similarly as any other portion of iterations with the same width. Although GENUAD and SMERED$^+$ seem to have this ideal behaviour, GENUAD mixes

better than SMERED$^+$. DIU has failed to pass this first convergence diagnostic. One question that needs to be addressed is why the trace plot for the DIU chain exhibits such a pattern, and this will be discussed later in this chapter. For now, the convergence diagnostics exclude the DIU algorithm because, as shown by its trace plot, the simulation has not converged to the stationary distribution.

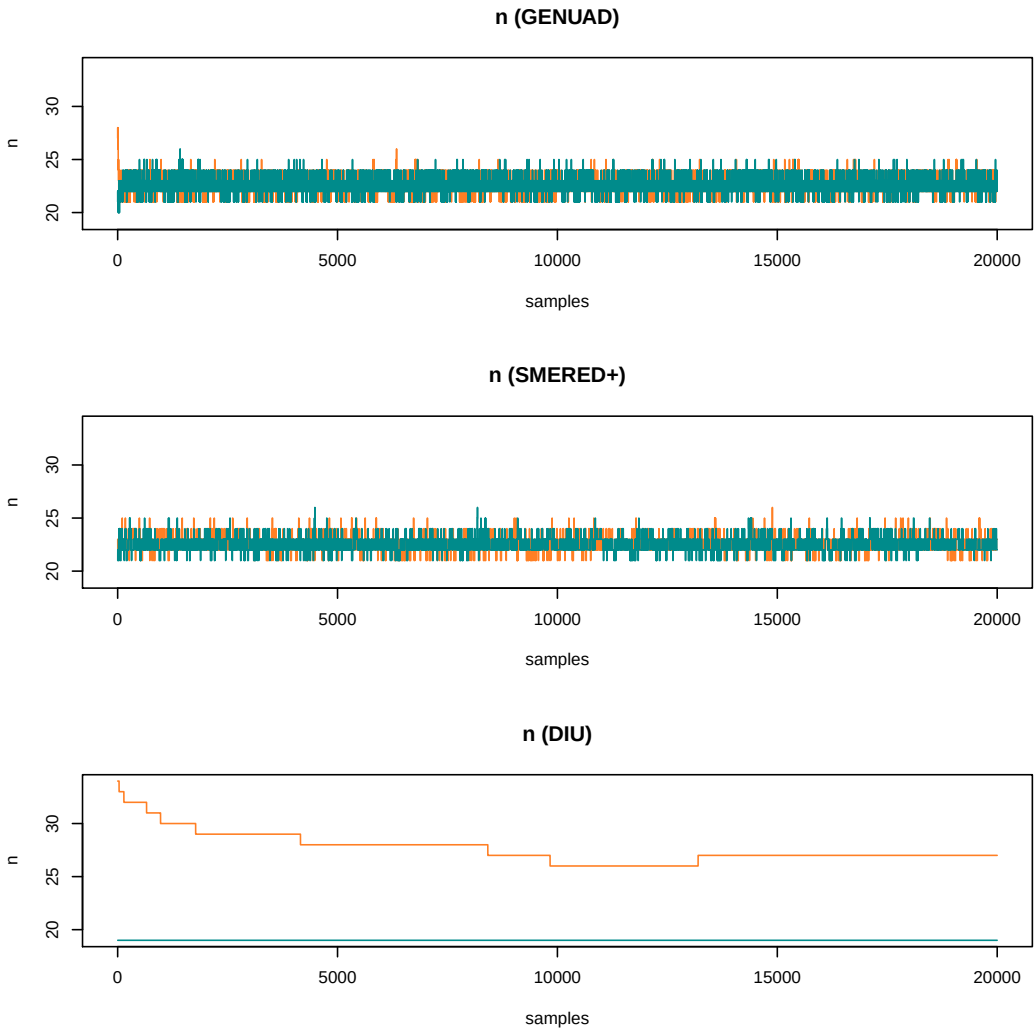


Figure 6.3: Two Markov chains for the GENUAD, SMERED$^+$, and DIU algorithms.

From Figure 6.3, the chains generated by GENUAD and SMERED$^+$ seem to converge quickly. To verify if the two chains are sampling from the same target distribution the Gelman and Rubin diagnostic presented in Section 2.3 was applied to these chains. The convergence is diagnosed when the chains forget their initial values, and they are indistinguishable. The diagnostic utilises the point estimates of the potential scale reduction factor (labelled Point est.) and their upper confidence limits (labelled Upper C.I.). Table 6.3 shows that these estimates are very close to 1 for both chains. Thus, there is no evidence to indicate a lack of convergence for the GENUAD and SMERED$^+$

chains.

Table 6.3: Gelman and Rubin diagnostic

Chain	Point est.	Upper C.I.
GENUAD	1.00128	1.001284
SMERED$^+$	1.00093	1.000944

Figure 6.4 details the different values of n by zooming in between the 100th and the 600th samples of the chains starting at $n = 34$ (the choice of the chain is arbitrary). This figure allows a better appreciation of the behaviour of the chains for exploring $\mathcal{X}_g$. It shows long-term entries at $n = 22$ and $n = 23$ for the SMERED$^+$ chain[1], and active motion between these two values for the GENUAD chain. Thus, from Figures 6.3 and 6.4, the GENUAD and SMERED$^+$ algorithms explore $\mathcal{X}_n$ in different ways. While SMERED$^+$ has long entries in the apparent modes of n, GENUAD fluently moves between them. This could be an indicator that GENUAD converges faster than SMERED$^+$.

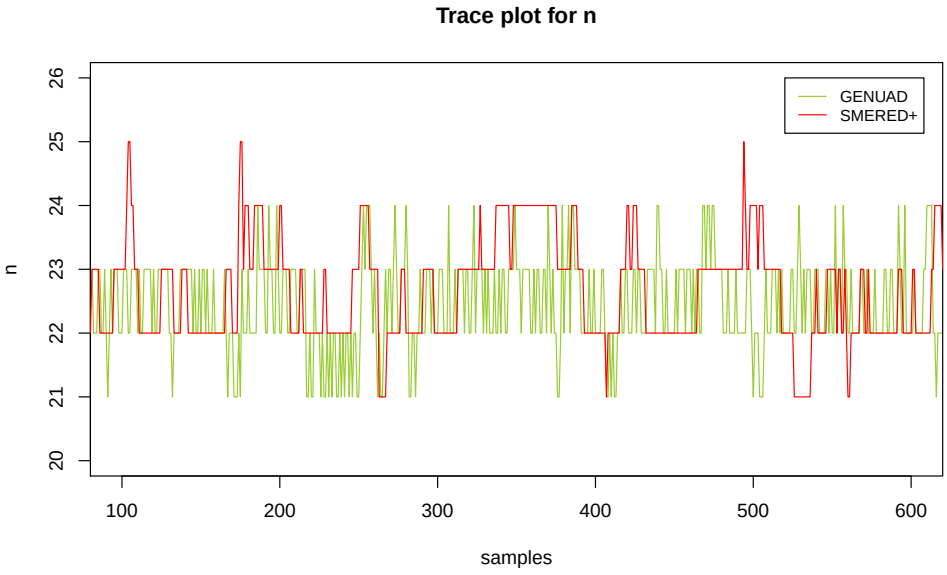


Figure 6.4: Zooming in one of the chains generated by GENUAD and SMERED$^+$ between the 100th and 600th samples.

The posterior distribution of n simulated by GENUAD and SMERED$^+$ are now compared. Figure 6.5 illustrates the probabilities associated with the posterior distribution of n, which are explicitly provided by Table 6.4. Interestingly, GENUAD and

[1]Recall that n is a discrete variable, which changes by up to one unit at each iteration of SMERED$^+$. Because the chain has been thinned, and the thinning factor is not large enough, there is a significant chance that two consecutive values of n are equal.

SMERED$^+$ seem to point to the same distribution as the probabilities are very close. However, these simulated distributions need further analysis to determine whether GENUAD and SMERED$^+$ are sampling values of n from the same distribution. The corresponding measures of central tendency and variability provide more information about the posterior distribution of n.

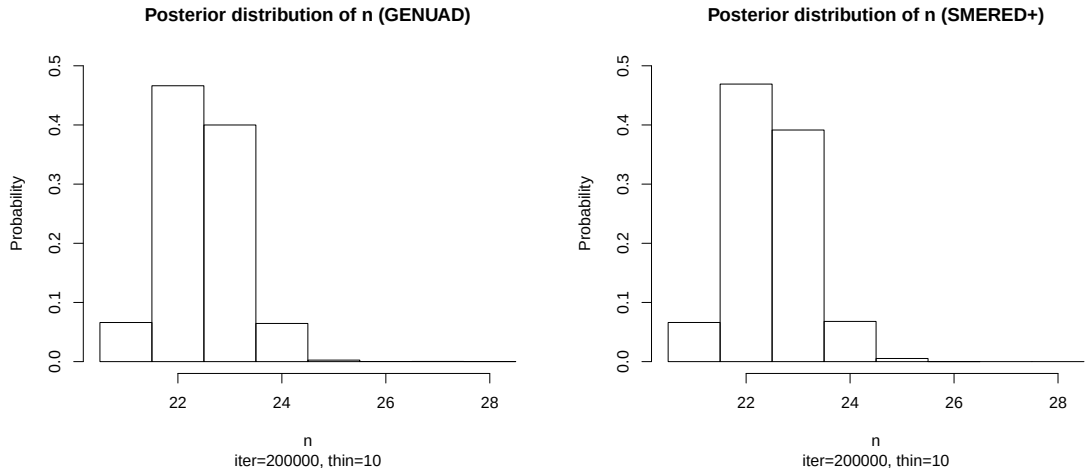


Figure 6.5: Posterior distribution of n simulated by GENUAD and SMERED$^+$.

Table 6.4: Posterior distribution of n

n	GENUAD	SMERED$^+$
21	0.06610	0.06615
22	0.46625	0.46910
23	0.39995	0.39145
24	0.06450	0.06795
25	0.00255	0.00530
26	0.00020	0.00005
27	0.00030	
28	0.00015	

The discrepancies of the two distributions in measures of central tendency, variability and shape can be easily identified by looking at the Q-Q plot, rather than the two histograms, side by side. If the points in the Q-Q plot tend to follow the line $y = x$ in a $xy-$plane, the two distributions can be considered as the same. Figure 6.6 shows the Q-Q plot for the quantiles of n in the two chains generated by both GENUAD (x-axis) and SMERED$^+$ (y-axis). Although Figure 6.5 displays similarities between the distributions, Figure 6.6 shows that the distribution in the case of SMERED$^+$ has a thinner upper tail. It goes to zero much faster than GENUAD, and so have less mass in the tail. So, the posterior distributions are slightly different. However, this may be associated with the fact that the GENUAD algorithm needs a burn-in period.

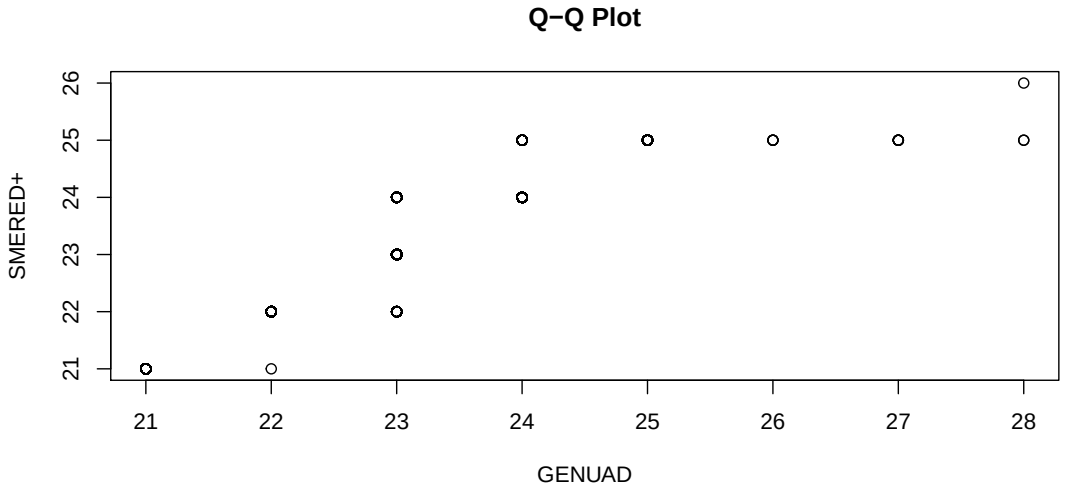


Figure 6.6: Quantile-Quantile plot of the posterior distributions of n simulated by GENUAD and SMERED$^+$.

The summary of the posterior distribution of n below shows measures of central tendency and variability. It is a modified output from the `CODA` library to show results for the two algorithms. While the first part displays the empirical mean of the sample, its standard deviation, and standard error estimates, the second part displays the quantiles. The empirical standard deviation estimates the square root of the variance of the posterior distribution of n. The precision of the empirical mean as a point estimate for the true posterior mean is measured by using the standard error. It depends on the number of iterations and the degree of autocorrelation within the sample. The output shows two estimates for such precision. The first estimate is the naive standard error, which is the standard deviation divided by the square root of the number of iterations, and it ignores the autocorrelation of the chain. The second estimate is the time-series estimate, which gives the asymptotic standard error, and it corrects the naive standard error for autocorrelation. For GENUAD and SMERED$^+$, the empirical means are very similar, 22.473 and 22.477, with standard deviations 0.734 and 0.741, respectively. Furthermore, their naive standard errors are small and similar.

```
Summary of the posterior distribution of n
Iterations = 1:200000
Thinning interval = 10
Number of chains = 1
Sample size per chain = 20000

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

 Chain         Mean       SD  Naive SE  Time-series SE
 GENUAD    22.47370  0.73399   0.00519         0.01279
 SMERED+   22.47730  0.74170   0.00525         0.02109
```

```
2. Quantiles for each variable:
```

Chain	2.5%	25%	50%	75%	97.5%
GENUAD	21	22	22	23	24
SMERED$^+$	21	22	22	23	24

The summary of the posterior distribution of n above does not indicate the convergence to the invariant distribution. However, when used with Figure 6.6 may provide an initial assessment of the similarity between the distributions simulated. Indeed, from the above discussion, there is substantial evidence to assert that the Markov chains generated by GENUAD and SMERED$^+$ converge to the same target distribution. Although they explore $\mathcal{X}_n$ very differently, the posterior distributions obtained show that they are sampling from the same posterior distribution of n.

Autocorrelation and effective sample size

If there is a significant correlation between neighbour samples in the chain, then it is possible that the simulated sample may be unable to reveal valuable information about the posterior distribution. Section 2.3 mentioned the lag autocorrelations as indicators of good and fast mixing of the chains. Namely when to stop the chain such that the sample obtained is representative of the target distribution. Also, it presented the definition of the effective sample size, and a strategy for estimating burn-in time, which indicates when the chain has begun to show the features of the target distribution.

Figure 6.7 shows the lag autocorrelations for the chains generated by GENUAD and SMERED$^+$ (when starting at $n = 34$). The figure at the top indicates that, in both cases, the autocorrelations are very weak. That is, both algorithms produce neighbour samples (nearby iterations) which are almost uncorrelated, signalling good and fast mixing. The dashed line marks a region that has been chosen to be focused in on, which results in the bottom figure. It detects small discrepancies between the autocorrelations.

The effective sample size (ESS) for GENUAD and SMERED$^+$ chains were computed by using the `CODA` library in the `R` software. They are equal to 3292.797 and 1236.788, respectively. The greater the number of independent samples, the better the efficiency of the algorithms. Thus, the ESS above indicate that GENUAD mixes faster than SMERED$^+$. Figure 6.8 shows the values of ESS for different lengths of the burn-in, when a maximum of 10 000 iterations has been fixed. The plot illustrates that it is unnecessary to discard samples from the GENUAD chain. Similarly, the maximum value of ESS is achieved when the burn-in period is equal to 1 for SMERED$^+$. However, it is not as efficient as GENUAD.

Additional diagnostics

The Geweke, Heidelberger-Welch, and Raftery-Lewis diagnostics were also used to assess convergence of GENUAD and SMERED$^+$ (by using the `CODA` library).

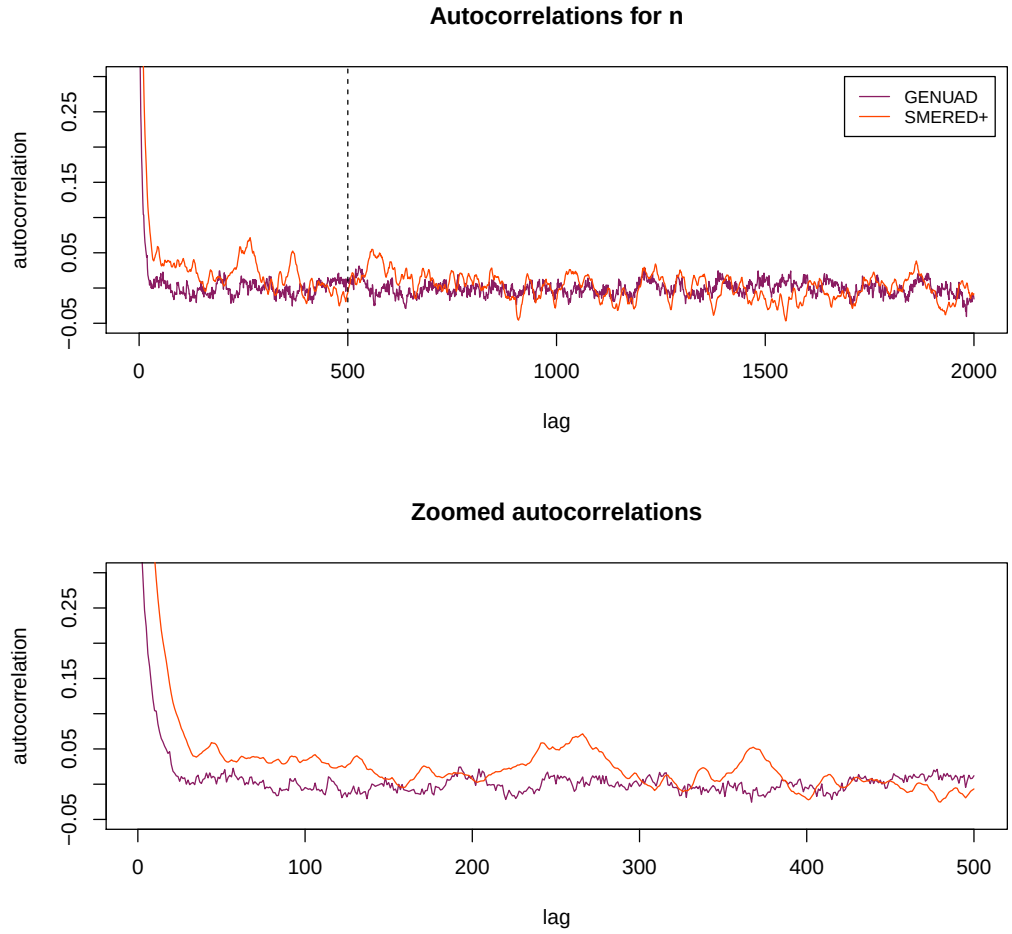


Figure 6.7: Autocorrelations for GENUAD and SMERED$^+$ for the data with $R = 2$.

Section 2.3 explained Geweke diagnostic, which compares the means of n obtained from two different "windows". For GENUAD, the Z-score with fractions in the first and second windows of 0.1 (the first 10%) and 0.5 (the last 50%) is 0.0483, while for SMERED$^+$ it is 1.7590. These values imply that for both chains, the samples in the two chosen windows come from the same distribution.

Table 6.5 shows the results of Heidelberger and Welch diagnostic where the target value for the ratio of half-width to sample mean has been defined as $\epsilon = 0.1$. The table indicates that the GENUAD and SMERED$^+$ chains have passed the stationarity and half-width tests. The former means that both chains are long enough to conclude that the sample has been drawn from the stationarity distribution. The latter indicates that the half-width is less than ϵ times the sample mean. Thus, the sample mean can be estimated with sufficient accuracy using the current length of the sample.

Section 2.3 has explained that the Raftery-Lewis diagnostic is based on the quantiles of the target distribution. For a specified quantile q to be estimated, the idea

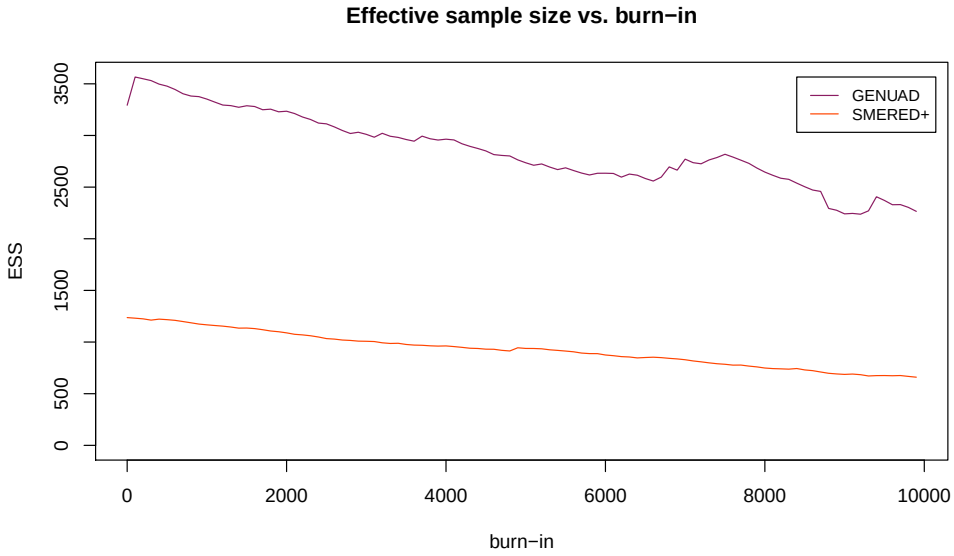


Figure 6.8: ESS against burn-in for GENUAD and SMERED$^+$ when $R = 2$.

Table 6.5: Heidelberger and Welch diagnostic for the data with $R = 2$.

	Stationarity			Halfwidth		
Chain	Test	Start iter	p-value	Test	Mean	Halfwidth
GENUAD	passed	1	0.210	passed	22.5	0.0251
SMERED$^+$	passed	1	0.366	passed	22.5	0.0413

is to estimate u such that $\Pr(n \leq u) = q$ with a precision of $r = 0.05$ and a probability of obtaining an estimate in the interval $(u - r, u + r)$ equal to $s = 0.95$. The precision required to estimate the time to convergence is 0.001. With these settings, the `CODA` output for this diagnostic is a set of estimates, namely, the length of burn-in (M), total of simulations (T), the minimum number of iterations based on zero autocorrelation ($T_{\min}$), and a dependence factor (I), which is defined as $I = (M+T)/T_{\min}$.

Brooks and Roberts (1998) suggest that when the interest is a set of quantiles, the user should take the largest of the resulting estimate burn-in lengths. Here the test has been applied to the quantiles $q \in \{0.1, 0.2, \ldots, 0.9\}$. The results are shown in Table 6.6. Taking the maximum, Table 6.6 indicates that the estimation of the set of quantiles of n to within ± 0.05 with 95% probability needs a minimum of 1856 samples from GENUAD and 4401 samples from SMERED$^+$. This also suggests that GENUAD reaches stationarity faster than SMERED$^+$. Both chains require short burn-in sample. The dependence factor (I) has small values for GENUAD. Since values of I larger than 5 indicate strong autocorrelation, the samples generated display high correlations for SMERED$^+$, as already concluded. Thus, it seems that the chains with a length of 20 000 generated by GENUAD and SMERED$^+$ simulate the posterior distribution of n with the desired precision.

Table 6.6: Raftery-Lewis diagnostic for the data with $R = 2$.

	Quantile	Burn-in (M)	Total (T)	Lower bound ($T_{\min}$)	Depend. factor (I)
GENUAD	0.1	12	1856	139	13.40
	0.2	12	1856	246	7.54
	0.3	12	1856	323	5.75
	0.4	12	1856	369	5.03
	0.5	12	1856	385	4.82
	0.6	12	456	369	1.24
	0.7	12	456	323	1.41
	0.8	12	456	246	1.85
	0.9	12	456	139	3.28
SMERED$^+$	0.1	36	4401	139	31.70
	0.2	36	4401	246	17.90
	0.3	36	4401	323	13.60
	0.4	36	4401	369	11.90
	0.5	36	4401	385	11.40
	0.6	18	482	369	1.31
	0.7	18	482	323	1.49
	0.8	18	482	246	1.96
	0.9	18	482	139	3.47

None of the diagnostics above detect a lack of convergence of the simulated chains. Therefore, the analysis of the trace plots, the autocorrelations, and the convergence diagnostics suggest strong evidence that both the GENUAD and SMERED$^+$ algorithms generated Markov chains, which are very close to the stationary distribution.

Transition probabilities of the simulated chains

The thinning factor of the Markov chains implies that the transition probabilities refer to shifts every 10 iterations (i.e. $\Pr(n^{(t+10)} = j | n^{(t)} = i)$). These probabilities were estimated using the frequency of moving from i to j for each chain. The difference between GENUAD and SMERED$^+$ is the focus here because DIU has shown inferior performance using this particular data. Figure 6.9 shows such differences where the order has been taken as $\Pr_{\text{SMERED}^+} - \Pr_{GENUAD}$. The blue colour indicates that the self-loops have greater probability in the SMERED$^+$ chain than in GENUAD. While GENUAD moves smoothly between different values of n, SMERED$^+$ displays a certain "laziness" for moving to a different state. This topic will be discussed later.

6.3 Two or more PCR replicates

Section 1.4 explained that Wright et al. (2009) considered two PCR replicates ($R = 2$) in the data provided by Frantz et al. (2003). However, the full data of Frantz et al.

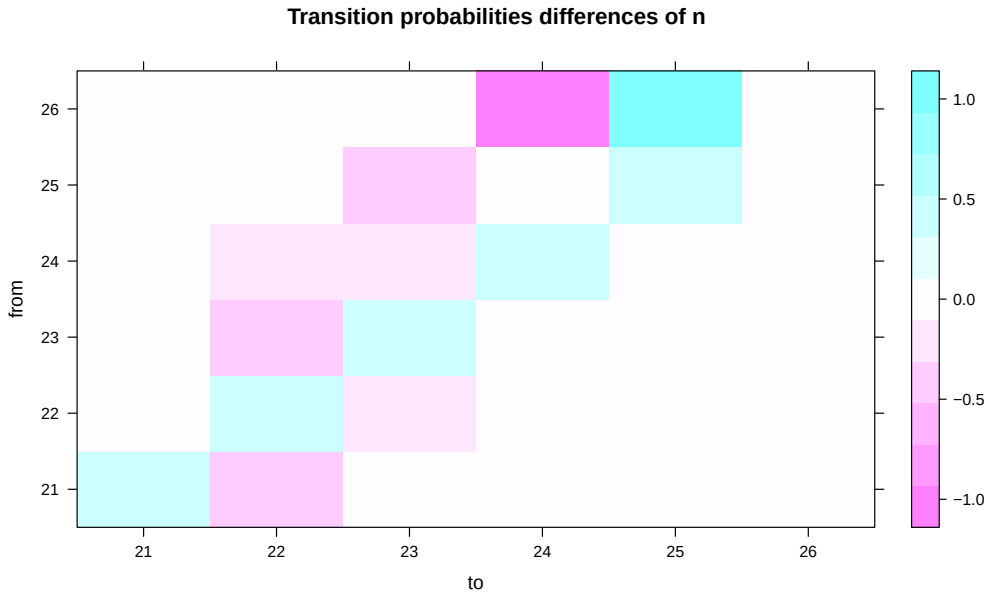


Figure 6.9: Differences between the transition probabilities in the state space of n of SMERED$^+$ and GENUAD for $R = 2$.

has two or more replicates at each loci. This section considers the full data, for which $R \geq 2$ and the loci do not have the same number of replicates. The consensus genotypes in the g^{obs} matrix and the number of alleles at each locus, denoted by m, change. Now, $m = (6, 4, 4, 4, 3, 4, 3)$ with $\eta = (21, 10, 10, 10, 6, 10, 6)$, the number of possible genotypes at each loci. The fixed values for N and p are the same as before, and the allele frequencies in γ are equally probable at each locus.

The trace plots in Figure 6.10 show that the chains generated by GENUAD meet after approximately 10 000 iterations of the algorithm. The Gelman diagnostic in Table 6.7 indicates that the chains generated by each algorithm are sampling from the same distribution, as the point estimates of the potential scale reduction factor are close to 1.0. Figure 6.11 shows that the distributions are similar.

Table 6.7: Gelman and Rubin diagnostic

Chain	Point est.	Upper C.I.
GENUAD	1.005250	1.025852
SMERED$^+$	1.002488	1.012436

The following is a summary of the posterior distribution of n for both chains. It shows similar measures of central tendency and variability of the distributions. In particular, the empirical means are similar with small standard deviations. As previously mentioned, this summary does not indicate the convergence to the invariant distribution. However, it provides information about the similarity between the distributions simulated.

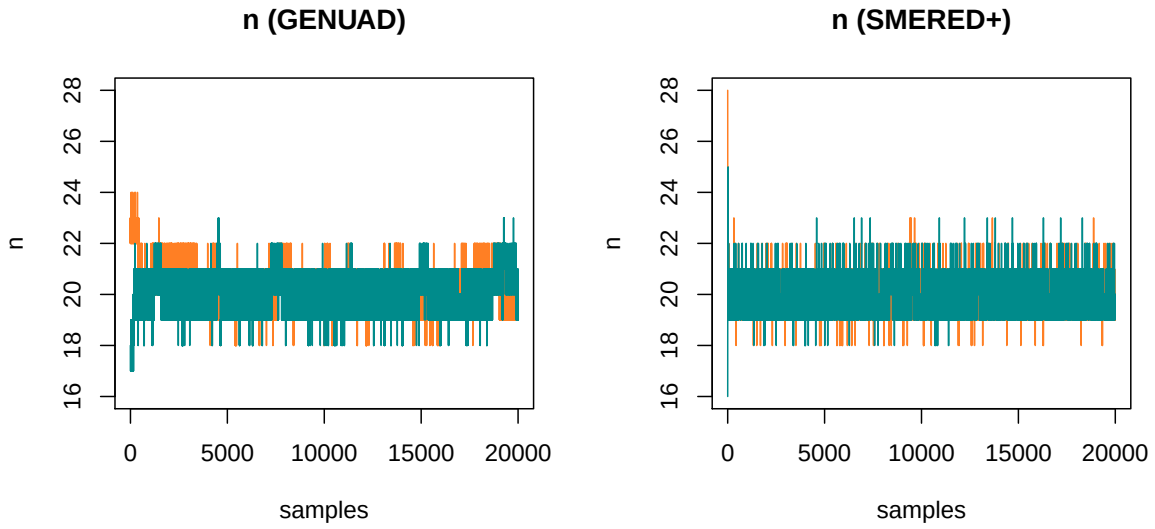


Figure 6.10: Trace plots of the chains generated by GENUAD and SMERED$^+$ for $R \geq 2$.

```
Summary of the posterior distribution of n
Iterations = 1:200000
Thinning interval = 10
Number of chains = 1
Sample size per chain = 20000

1. Empirical mean and standard deviation for each variable,
plus standard error of the mean:

 Chain          Mean        SD  Naive SE  Time-series SE
 GENUAD    19.957750  0.837974  0.005925        0.101137
 SMERED+   19.813200  0.788122  0.005573        0.019907

2. Quantiles for each variable:

 Chain     2.5%  25%  50%  75%  97.5%
 GENUAD      19   19   20   20     22
 SMERED+     19   19   20   20     21
```

Figure 6.12 shows the lag autocorrelations for the chains generated by GENUAD and SMERED$^+$ for the data with two or more replicates. It shows a significant change in the autocorrelation for GENUAD, which was very weak in the two replicates data. For both GENUAD and SMERED$^+$ algorithms, the autocorrelations of neighbour values of n in the simulations is near 0.5. The difference is that for SMERED$^+$ the autocorrelations decay quickly than for GENUAD. The GENUAD chain should be lagged for more than 500 values of n (taking the thinning into account, it would be equivalent to 5000 iterations of the algorithm) to have values of n with correlations lower than 0.2. These correlations are not excessively high, but they are very unstable.

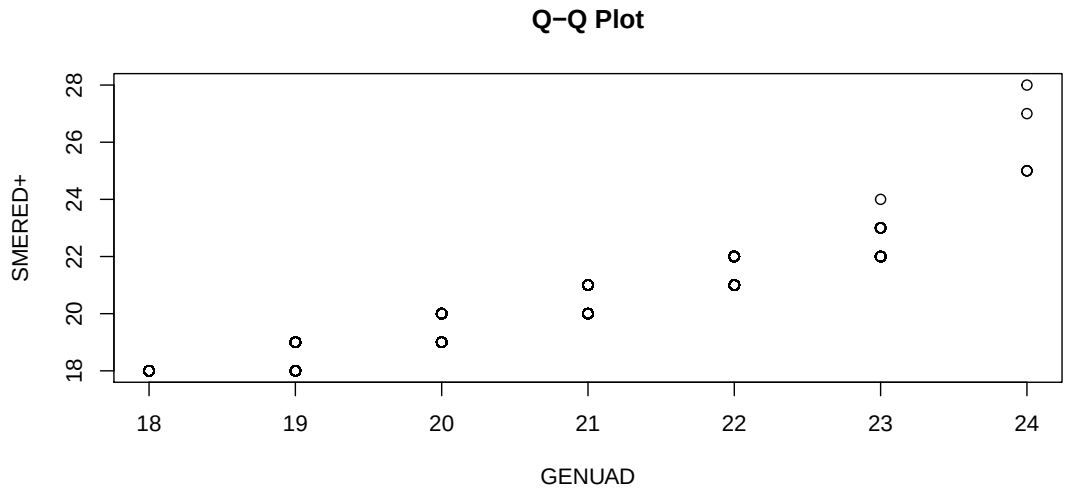


Figure 6.11: Quantile-Quantile plot of the posterior distributions of n simulated by GENUAD and SMERED$^+$ for $R \geq 2$.

This behaviour may be a sign of poor mixing in GENUAD. In contrast, the autocorrelations of SMERED$^+$ display similar behaviour in both cases $R = 2$ and $R \geq 2$, with weak and steady correlations.

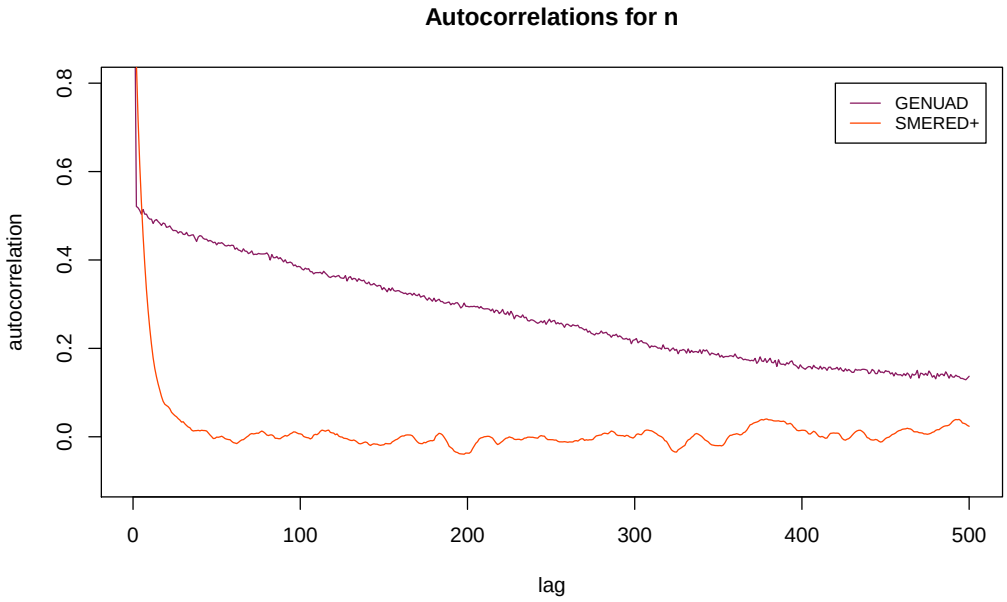


Figure 6.12: Autocorrelations for GENUAD and SMERED$^+$ for the data with $R \geq 2$.

The ESS are 68.65 and 1567.42 for the GENUAD and SMERED$^+$ chains, respectively. When compared with the case of two replicates, the ESS for GENUAD dras-

tically changes (it drops from 3292.797 to 68.65), while for SMERED$^+$ the change is less severe (it increases from 1236.788 to 1567.42). With the aim of comparing these changes for the distinct the data sets, Figure 6.13 shows the values of the ESS for different values of the burn-in for the data with two replicates and the data with two or more replicates. The solid lines are used for the case when $R \geq 2$, while the dashed lines are the same as in Figure 6.8 for $R = 2$. While there is an extreme change for GENUAD, the SMERED$^+$ algorithm remains almost unaltered.

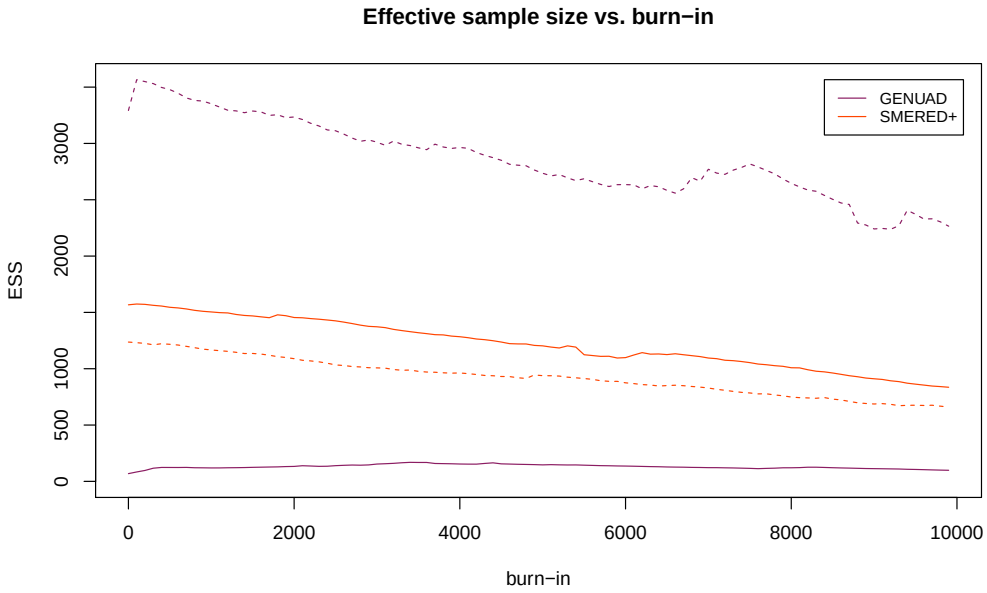


Figure 6.13: Solid lines represent the values for $R \geq 2$, and dashed lines for $R = 2$.

Figure 6.13 illustrates the steady ESS values of SMERED$^+$. Although these ESS values may not be as high as those shown by GENUAD in the case of two replicates, they are consistent. Possible reasons explaining this behaviour will be discussed. Lastly, Figure 6.14 shows the transition probabilities, which have a similar pattern to those in Figure 6.9. However, the differences are smaller in magnitude.

Additional diagnostics

The results of applying Geweke, Heidelberger-Welch, and Raftery-Lewis diagnostics for the case $R \geq 2$ are as follows. The Z-score of Geweke diagnostic for GENUAD, with fractions in the first and second windows of 0.1 and 0.5, is 0.9281. For SMERED$^+$, it is -0.6773. For both chains, the samples in the two chosen windows come from the same distribution. Table 6.8 indicates that both chains are long enough to conclude stationarity, however, GENUAD starts to show features of the stationary distribution after 2000 iterations. Additionally, the sample mean can be estimated with sufficient accuracy ($\epsilon = 0.1$) using the current length of the sample.

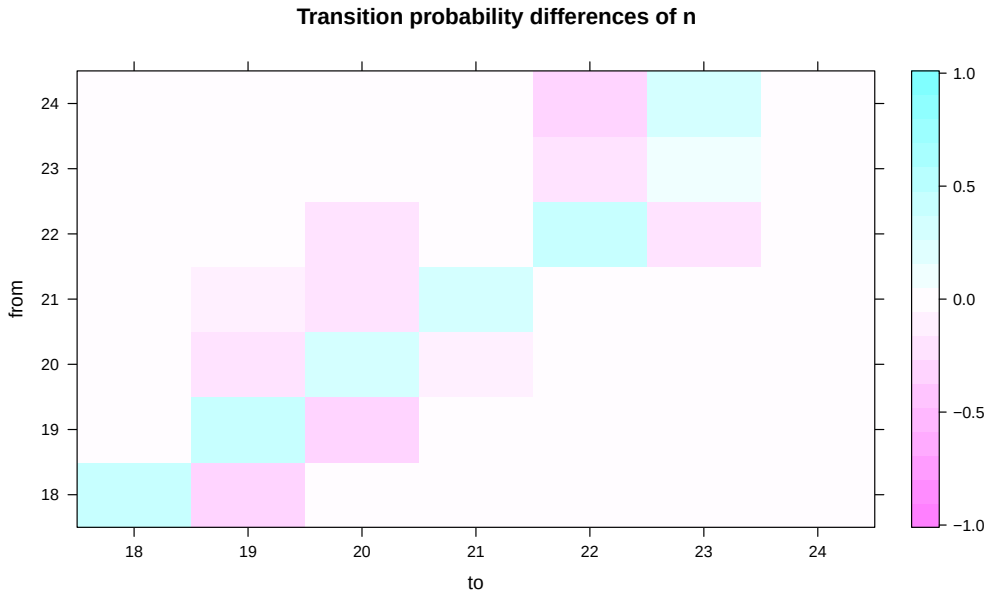


Figure 6.14: Differences between the transition probabilities in the state space of n of SMERED$^+$ and GENUAD for $R \geq 2$.

Table 6.8: Heidelberger and Welch diagnostic for the data with $R \geq 2$.

	Stationarity			Halfwidth		
Chain	Test	Start iter	p-value	Test	Mean	Halfwidth
GENUAD	passed	2001	0.198	passed	19.9	0.127
SMERED$^+$	passed	1	0.318	passed	19.8	0.039

With the previous settings ($s = 0.95$, $r = 0.05$), Table 6.9 displays that to estimate the set of quantiles of n to within ± 0.05 with 95% probability, a minimum of 19040 samples of GENUAD and 3868 samples of SMERED$^+$ are needed. These results turn the status of the algorithms because now SMERED$^+$ reaches stationarity faster than GENUAD. Likewise, the dependence factor (I) has small values in the case of SMERED$^+$. Values of I larger than 5 indicate strong autocorrelation in the chain generated by GENUAD, as previously concluded. Thus, although both chains with the length of 20 000 reach convergence, SMERED$^+$ converges faster and mixes better than GENUAD.

As previously illustrated, the analysis of the trace plots, the autocorrelations, and the convergence diagnostics strongly suggest that the chains generated by GENUAD and SMERED$^+$ converge to the stationary distribution.

Table 6.9: Raftery-Lewis diagnostic for the data with $R \geq 2$.

	Quantile	Burn-in (M)	Total (T)	Lower bound ($T_{\min}$)	Depend. factor (I)
GENUAD	0.1	156	16887	139	121.0
	0.2	156	16887	246	68.6
	0.3	156	16887	323	52.3
	0.4	238	19040	369	51.6
	0.5	238	19040	385	49.5
	0.6	238	19040	369	51.6
	0.7	238	19040	323	58.9
	0.8	612	6528	246	26.5
	0.9	612	6528	139	47.0
SMERED$^+$	0.1	36	3868	139	27.80
	0.2	36	3868	246	15.70
	0.3	36	3868	323	12.00
	0.4	24	1560	369	4.23
	0.5	24	1560	385	4.05
	0.6	24	1560	369	4.23
	0.7	24	1560	323	4.83
	0.8	24	1560	246	6.34
	0.9	20	195	139	1.40

6.4 Summary

The three Markov chains have different performances when considering the data used. For the toy example of badger genotypes, they all accurately simulate the target distribution of g in Eq. (3.10). For the large datasets, only SMERED$^+$ and GENUAD manage to simulate the posterior distribution of n. The diagnostics applied to these algorithms showed that there is strong evidence to conclude the convergence of the chains. In general, the simulations show two different chains sampling from the same distribution, but evolving in distinct ways.

The GENUAD algorithm seems to have better performance when $R = 2$ than SMERED$^+$. When comparing the generated chains, the GENUAD autocorrelations are smaller, and the effective sample size is more significant. The Raftery-Lewis diagnostic suggests that, for reaching stationarity, SMERED$^+$ needs to double the number of iterations that GENUAD requires. However, when $R \geq 2$ the SMERED$^+$ algorithm is favoured. The autocorrelations for GENUAD unsteadily increased and the effective sample size substantially reduced. In this case, the Raftery-Lewis diagnostic suggested that SMERED$^+$ reaches stationarity faster than GENUAD. Remarkably, this does not mean that the number of iterations needed for SMERED$^+$ has decreased with respect to the case when $R = 2$. Instead, it means a considerable increase in the number of iterations required for GENUAD, while the results for SMERED$^+$ remains virtually unaltered. This topic is discussed more extensively in the next chapter.

Regarding the DIU algorithm, although it succeeded in the toy example, it failed for the other two examples. Figure 6.3 shows that the DIU algorithm generates a Markov chain that gets stuck at a particular value of n. Because of the DIU strategy of updating a single row of g, two consecutive iterations will either give equal values of n or values that may differ by one unit. However, this strategy alone does not cause the extreme slow motion of DIU through the values of n observed in Figure 6.3. The fact that SMERED$^+$ also generates values of n such that two consecutive iterations may differ by one unit, at the most, supports the previous statement. However, as shown by Figure 6.3, the behaviour of the SMERED$^+$ and DIU chains are very different. The following chapter will discuss possible explanation for the pattern of the DIU trace plot.

Chapter 7

Discussion

The previous chapter presented results from implementing the GENUAD, SMERED$^+$, and DIU algorithms in three different cases: a small-scale example and two larger datasets. Although the GENUAD and SMERED$^+$ algorithms correctly simulated the posterior distribution of interest, they exhibited significant differences when applied to the largest datasets. To explore this further, this chapter is divided into two parts. First, it discusses apparent discrepancies when simulating the posterior distribution in Eq. (3.10) and explains their causes. It addresses questions such as which algorithm could be potentially better for simulating such distribution, and under which circumstances. Second, the chapter discusses situations which may weaken or strengthen the algorithms. In particular, it examines the effect that extreme values of the fixed parameters N, γ, and p could have on the performance of the algorithms. It may shed some light on possible solutions for improving the sampler.

7.1 GENUAD vs. SMERED$^+$

The simulations in the previous chapter not only revealed features of the algorithms but also gave rise to several questions. For instance, when $R \geq 2$, why is SMERED$^+$ favoured? In addition, are there specific circumstances in which GENUAD performs better than SMERED$^+$? Finally, what are the implications of the different behaviours of the chains for traversing the state space of interest? This section aims to explore possible answers to these questions. The results of the simulations in the previous chapter are crucial for such discussion.

Number of replicates and mixing of the chains

The use of two different datasets for the badger genotypes revealed some features of GENUAD. Its performance in the case of two PCR replicates ($R = 2$) is preferred to SMERED$^+$. In comparison, its performance is inferior to SMERED$^+$ when applied to the case of two or more replicates ($R \geq 2$). The differences between the two data sets can explain the reasons for these distinctive results.

When $R = 2$ there is a higher degree of degraded or contaminated data than when

$R \geq 2$. This contamination refers to the number of observed homozygotes in the $S = 47$ DNA samples, which is higher when $R = 2$. According to Wright (2011, p. 46), "where the results from these replicates disagree the rule of thumb currently used is that the result containing the maximum number of alleles is accepted". This assertion means that, at a particular locus, when deciding between a homozygous and heterozygous genotype (given that one of the alleles in the putative heterozygote matches the allele in the homozygote), the heterozygous genotype wins. The more replicates are performed, the higher the chance of assigning heterozygous genotypes. Thus, the homozygosity in g^{obs} reflects the degree of contamination in the data. So, the question is why does the presence of heterozygotes in the observed sample more strongly influence the performance of GENUAD?

As outlined in Section 3.2.1, the GENUAD algorithm is a Gibbs sampler which alternately updates $\mathcal{G}$ and X. The problem is that the sampler may struggle to move if these parameters are strongly correlated. A significant correlation between $\mathcal{G}$ and X takes place when the data is crowded with heterozygous genotypes which do not allow enough freedom for moving in the state space $\mathcal{X}_n$, conditioned on a state of $\mathcal{G}$. To explain this point consider the following example.

Example 7.1.1. Consider g^{obs} as below, with two alleles at each locus.

$$g^{\text{obs}} = \begin{pmatrix} 1,2 & 1,1 \\ 1,1 & 1,1 \\ 1,2 & 1,2 \end{pmatrix}.$$

Suppose that the current states for $\mathcal{G}$ and X are given by

$$\mathcal{G}^{(t)} = \begin{pmatrix} 1,2 & 1,2 \\ 1,1 & 2,2 \\ 2,2 & 2,2 \end{pmatrix} \quad \text{and} \quad X^{(t)} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

When using GENUAD, if the following step is to update X given $\mathcal{G}$, X cannot be changed. This is because, for each observed genotype, there are no other compatible genotypes more than those currently assigned in $\mathcal{G}^{(t)}$. In other words, the unique possible move for X is $X^{(t+1)} = X^{(t)}$, for which $n^{(t)} = 1$ stays unaltered with probability 1.0. However, X may change once $\mathcal{G}$ is updated.

In contrast, SMERED$^+$ can easily update the X matrix above, since $\mathcal{G}$ and X are simultaneously updated. Suppose that samples 1 and 2 are randomly chosen to be updated. They are proposed for splitting because they are currently linked to the same individual. The following proposal $(\mathcal{G}^*, X^*)$ with $n^* = 2$ can be accepted with positive probability.

$$\mathcal{G}^* = \begin{pmatrix} 1,2 & 1,1 \\ 1,2 & 1,2 \\ 1,1 & 1,1 \end{pmatrix} \quad \text{and} \quad X^* = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

□

Example 7.1.2. Let $g^{\text{obs}} = \begin{pmatrix} 1,1 \\ 1,2 \end{pmatrix}$ be the observed genotypes at a single locus with two alleles. In this case, $\mathcal{X}_g$ has two elements given by

$$\mathcal{X}_g = \left\{ \begin{pmatrix} 1,2 \\ 1,2 \end{pmatrix}, \begin{pmatrix} 1,1 \\ 1,2 \end{pmatrix} \right\}, \tag{7.1}$$

for which $n = 1$ and $n = 2$, respectively. Figure 7.1 shows $\text{supp}(f_{\mathcal{G},X})$. Note that when using GENUAD, if X is fixed, updating $\mathcal{G}$ will not update n. The value of n remains unchanged because all points have the same colour for a fixed ordinate. However, once X is updated, n may change, unless the current of $\mathcal{G}$ is $\mathcal{G}^i$ with $i = \{14, 15, 17, 18, 23, 24, 26\}$. For example, the three red points for $\mathcal{G}^{14}$ with ordinates X^1, X^5 and X^9 illustrate the situation where updating X does not change the value of n. The value of n does not change because there are no black points in that abscissa. Once G is updated, shifting to a different state that connects with a black point is possible. For example, moving from $(\mathcal{G}^{14}, X^1)$ to $(\mathcal{G}^{20}, X^1)$. □

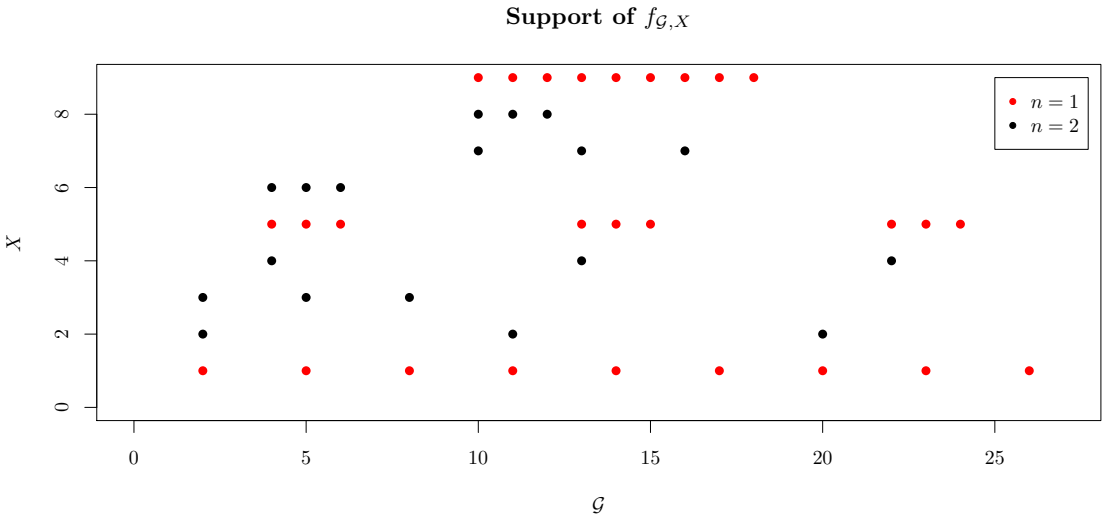


Figure 7.1: For g^{obs} as in

This example shows that X and $\mathcal{G}$ can be strongly correlated. This important correlation creates bottlenecks for exploring the state space of n, $\mathcal{X}_n$, because it may disrupt the generation of a new value of n. Also, high heterozygosity (data moderately contaminated) in g^{obs} means a strong correlation between $\mathcal{G}$ and X, while high homozygosity (data extremely contaminated) manifests a weak correlation. It is well known that strong correlations among the parameters imply high autocorrelations in the chain because the Gibbs sampler algorithm is based on full conditionals. Figures 6.12 and 6.13 evidenced this fact for GENUAD.

Therefore, both high heterozygosity in g^{obs} and the strategy of updating $\mathcal{G}$ and X by using a Gibbs sampler may impede the smooth exploration of the state space $\mathcal{X}_n$

when implementing the GENUAD algorithm. In contrast, the SMERED$^+$ algorithm conveniently updates $\mathcal{G}$ and X because it happens simultaneously and unconditionally. Thus, the inclusion of more than two replicates in the data is in detrimental to the performance of GENUAD when generating almost uncorrelated values of n.

Under what conditions should the GENUAD vs. SMERED$^+$ algorithms be used?

The answer to the question about which of the two algorithms is better is that it depends. According to the results above, the quality of the data and how long the user is willing to run the simulations are factors influencing which algorithm will more accurately simulate the posterior distribution of interest.

With regard to the quality of the data, the GENUAD algorithm worked well when the data had a reasonable number of homozygous genotypes. This homozygosity gives to the Gibbs sampler more freedom to explore the state space of n, $\mathcal{X}_n$, because of the weak correlation between $\mathcal{G}$ and X. It can produce a considerable amount of independent samples as they are directly generated from the target distribution, implying this that the Markov chain generated by GENUAD mixed very well. Thus, if the parameters of interest exhibit a weak correlation, GENUAD would be an efficient sampler for generating samples with low autocorrelations.

On the other hand, the corruption intensity of the data does not influence the SMERED$^+$ algorithm, as the dotted and dashed red lines in Figure 6.13 displayed. The SMERED$^+$ sampler generated approximately the same amount of samples with low autocorrelation, in both cases highly and weakly correlated parameters. Thus, SMERED$^+$ may be more attractive than GENUAD if there is no available information about the correlation between the parameters. In cases where this correlation is closely associated with data degradation, as in the genetic profiles, this feature of SMERED$^+$ is a remarkable advantage. That being said, it is often not possible to know how much distortion the data has. Even more, the presence of distorted data may not be recognisable.

Therefore, the recommendation is to implement the SMERED$^+$ algorithm in situations where the correlation between the parameters of interests is unavailable, and applying GENUAD if there is certainty of a weak correlation.

In addition to the quality of the data, whether the GENUAD vs. SMERED$^+$ algorithm should be used also depends on the number of simulations that the user is willing to run. This is related to the laziness that SMERED$^+$ exhibits. In simple words, a lazy chain is one that has a probability of at least $1/2$ to stay in the current state. The zoomed trace plots in Figure 6.4 and the transition probabilities in Figures 6.9 and 6.14 suggested that the Markov chain generated by SMERED$^+$ could be lazy. This laziness may delay convergence to the invariant distribution. Thus, the chain generated by the GENUAD algorithm may show features of the target distribution faster than SMERED$^+$. The following section explores other implications of this laziness.

Is laziness good or bad?

The laziness of a chain is a subject of discussion in the literature associated with Markov chains and mixing times literature. See Jerrum (2003), Levin et al. (2009), and Basu et al. (2017). Given an arbitrary transition matrix P of a Markov chain with size state space q, a *lazy* version of the chain is obtained by defining a transition matrix $Q = \frac{1}{2}(I_q + P)$ where I_q is the identity matrix of order q. In other words, for moving in the state space by using Q, the outcome of flipping a fair coin governs how the chain moves. If heads, it takes a step in P; otherwise, it stays in the current state. Producing a lazy Markov chain will fix problems regarding periodicity because as seen, lazy chains are aperiodic. Jerrum (2003) presents several remarks about lazy chains. First, the ergodicity of a Markov chain is transferred to its lazy version, in which case both converge to the same stationary distribution. In fact, the irreducibility of the original Markov chain is enough to prove ergodicity of the lazy version, since aperiodicity is trivial. Second, laziness has the effect of doubling the mixing time, that is, it slows down the original chain by a factor of two. However, all eigenvalues of the transition matrix of the lazy version are non-negative. This "avoids possible parity conditions that would lead to the Markov chain being periodic or nearly so", as stated in Jerrum (2003, p. 53), but it may also improve the bounds for mixing. As, explained by the authors, in the implementation of a lazy chain, "efficiency would not be compromised by laziness".

Figures 6.9 and 6.14 illustrated the differences between the transition probabilities of n for both chains in both cases $R = 2$ and $R \geq 2$. The blue colour in the diagonal suggested that the chain generated by SMERED$^+$ is lazy in both cases. This laziness is also observable from Figure 6.4, where the chain has long visits to the same value of n. This figure may also suggest a certain periodic behaviour of the GENUAD chain when moving between sets of n values. While the lazy chain stays inactive in a specific value of n and eventually moves to a different value, the GENUAD chain switches between $n = 22$ and $n = 23$. Figure 6.4 illustrates this pattern in only 500 iterations. For the entire simulated values, this tiny portion of all values simulated of n resembles a rectangle with a few lines up/down. Figure 7.2 (extracted from Figure 6.10) shows several of these rectangles which may suggest periodic behaviour of the GENUAD chain when exploring subsets of the state space. SMERED$^+$ removes periodicity problems.

Although this topic of lazy chains needs in-depth exploration and understanding, the results of the simulations show that laziness may not be a sign of weakness. It solves periodicity problems by increasing the probabilities of self-loops with no sacrifice of algorithm efficiency. These kinds of chains should be named *smart-lazy* chains rather than merely lazy chains. Metaphorically speaking, they follow the "path of least resistance". Rather than continually switching between the same two states, a smart-lazy chain stays longer at one state, and later it travels to the other one. The word "smart" in *smart-lazy* means that the chain is not generating values of n that will eventually have strong correlations (as GENUAD does). Although the chain may not generate a large number of values with low correlations, it will converge to the invariant distribution by simulating enough of them.

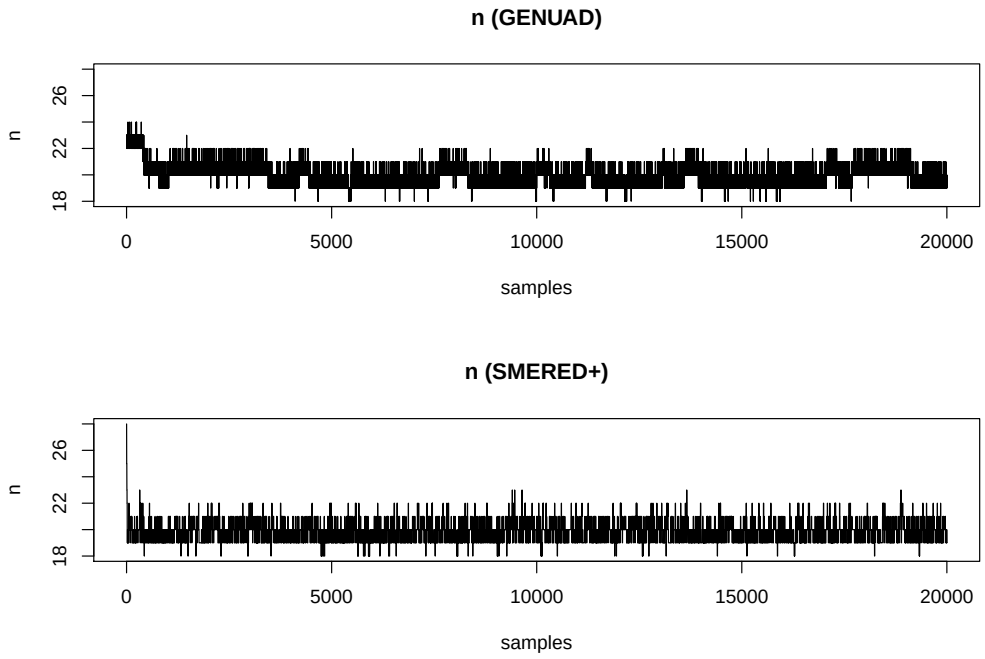


Figure 7.2: Trace plots for GENUAD and SMERED$^+$ when $R \geq 2$.

To summarise, the GENUAD and SMERED$^+$ algorithms explore $\mathcal{X}_n$ differently, but both gather the same information about n. GENUAD moves very efficiently within subsets of $\mathcal{X}_n$, but not between subsets. That is, after thoroughly exploring a subset, the sampler occasionally shifts to another which is also thoroughly explored before leaving. Thus, when exploring the state space, GENUAD could be locally efficient. On the other hand, SMERED$^+$ moves freely between all states, without getting stuck in a subset. However, it may remain slightly longer in the same state without moving at all. Nevertheless, SMERED$^+$ effortlessly explores the state space. The following section gives an interpretation of Figure 7.2 using an analogy.

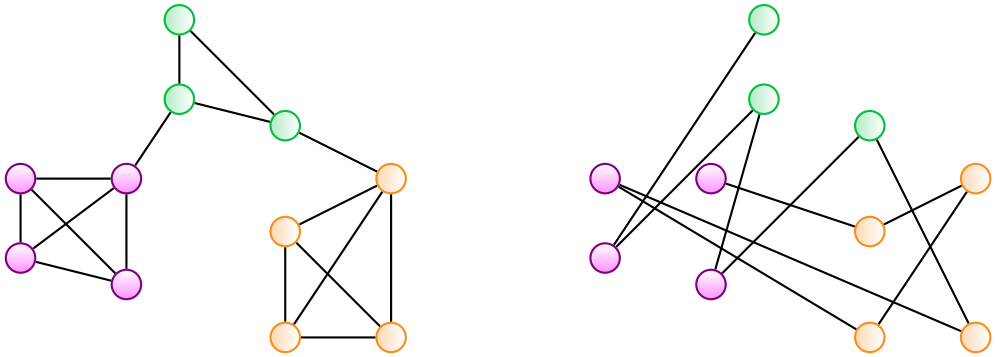
Two tourists exploring a city

Planning on how to get to know a new city is a matter of strategy and time. While some tourists prefer exploring new places by using GPS systems on their phones, other more adventurous travelers may enjoy wandering off track. Both options have their advantages and disadvantages, and there is a vast list of blogs with recommendations for different kinds of tourists.

For the purpose here, imagine that the GENUAD and SMERED$^+$ algorithms are two solo travellers in the same new city. GENUAD prefers to explore a neighbourhood in detail before visiting another one. SMERED$^+$ is determined to know what the whole town has to offer, and decides to visit different neighbourhoods without thoroughly knowing each one. These are the strategies behind Figure 7.2 for the case of $R \geq 2$. The question is which strategy is best for exploring the city? This ques-

tion is challenging and perhaps without a universal answer. However, one contributing factor is how much time is available. If the visit is for a short period of time, then the SMERED$^+$ strategy may be more effective because more knowledge about the city as a whole will be acquired. The knowledge of GENUAD would be limited to whatever neighbourhoods were visited in detail. For an extended visit, GENUAD and SMERED$^+$ will acquire the same knowledge.

Figure 7.3 displays both strategies. They resemble the city with three neighbourhoods identified by different colours. The nodes are tourist places, and the edges indicate if the solo travellers repeatedly travelled back and forward between the sights.



(a) (b)

Figure 7.3: (a) GENUAD as solo traveller exploring the city by neighbourhoods. (b) SMERED$^+$ as solo traveller exploring the city as a whole.

This analogy illustrates the situation suggested by the trace plot in Figure 7.2, where the neighbourhoods represent groups of values of n and the edges are shifts between them. For the third application (i.e. with $R \geq 2$ PCR replicates), GENUAD seems to be very efficient exploring subsets of the state space of $\mathcal{X}_n$, but it struggles for moving between subsets. Table 6.9 showed that the GENUAD requires more iterations, than SMERED$^+$. That is, more time to have enough knowledge about $\mathcal{X}_n$, the city.

7.2 Influence of the fixed parameters

For deriving Eq. (1.2), the parameters N, γ, and p were fixed. Indeed, the choice of these parameters beforehand has a significant impact on the performance of the MCMC algorithms studied here. This section examines how extreme values of the fixed parameters may affect the samplers for simulating the joint density of G and X.

Large populations

The population size N is a parameter that has a significant effect on SMERED$^+$. The definition of the Metropolis ratio in Eq. (5.3) depends on the ratio in Eq. (5.2), which involves N. If N is large, then:

- $N - n$ tends to increase the Metropolis ratio in Eq. (5.3). That is, there is a tendency to accept the split proposals.
- $(N - n + 1)^{-1}$ tends to decrease the Metropolis ratio in Eq. (5.3). That is, the merge proposals are prone to be rejected.

Large values of N favour splitting operations over merging operations. An interpretation of this is explained by using a specific situation. For example, consider two records with attributes in name, age, and major given by (Matthew Smith, 25, Statistics) and (Matt Smith, 27, Mathematics). Suppose that the population comprises all students attending university in New Zealand (large population size). When using SMERED$^+$, there are two cases:

i. If the two records are currently assigned to the same student, and a split is proposed. In this case, there is a high probability of splitting the records. This is because, given a large N, there would be a greater chance to have two students with similar attributes but are indeed different individuals.

ii. If the two records are currently assigned to different students, and merge is proposed. In this case, there is a low probability of merging the records for the same reasoning as above. That is, the large population of students results in a high probability that two records with similar information refer to two different students.

The situation would be different if the population was smaller, for example, only the students at the University of Otago (Dunedin, New Zealand). In this case, merging the records will not have a low probability of acceptance because it is likely that the two different records refer to the same student.

On the other hand, Eq. (5.7) aids in determining the effect of large values of N in DIU, for fixed values of γ. Indeed, $r \to \infty$ as $N \to \infty$. That is, for large values of N, if the proposal is a value of n greater than the current, then there is a high probability of acceptance. Nevertheless, if the proposal is a value of n smaller than the current, and N is large, then there is a high probability of rejection, as $r \to 0$ as $N \to \infty$.

Large values of N benefit GENUAD as $N - n$, the size of G^{mis}, changes. As shown in the proof of Theorem 4.1.1, G^{mis} plays an important role for exploring the support of the joint density of $\mathcal{G}$ and X.

Low allelic dropout probabilities

This section endeavours to determine how the allelic dropout probabilities p influence the algorithms. The parameter p is involved in the likelihood function given in Eq. (1.1), which includes the corruption process in the data. Section A.2.1 provides a definition for this probability. The full conditionals densities of $\mathcal{G}$ and X in GENUAD incorporate this term, and also the Metropolis ratio in SMERED$^+$. The Metropolis ratio in DIU is insusceptible to changes in the values of the dropout probabilities p

because the proposal distribution cancels the term $f(g^{\text{obs}}|g,p)$.

Following Section A.2.1, a small value of p in a single locus indicates a low probability of allelic dropout. This means that,

$$\Pr(\text{observed heterozygous} \mid \text{true heterozygous}) \to 1$$

$$\Pr(\text{observed homozygous} \mid \text{true heterozygous}) \to 0$$

For GENUAD, the first factor in Eq. (3.1) includes these probabilities. Observed homozygotes will link with true homozygotes with high probability, and with true heterozygotes with low probability, while observed heterozygotes may correspond to true heterozygotes with a probability close to 1.0. Thus, the updater of $\mathcal{G}$ tends to sample heterozygous genotypes with high probability. It is complicated to determine the impact of the first factor in Eq. (3.6) when p is small.

For SMERED$^+$, how small values of p alter the algorithm is also obscure. The definition of the Metropolis ratio is complicated, and the term appears in several parts of the ratio. Thus, studying the effect of small allelic dropout probabilities is challenging in GENUAD and SMERED$^+$ cases. It is a matter that needs further analysis.

High number of alleles and loci

Figure 6.3 showed that the DIU algorithm gets stuck in values of n. Chapter 6 continued without considering the algorithm. However, this section will explain that behaviour, and discuss possible approaches that could solve the problem.

Eq. (5.7) shows that the allele frequencies contained in γ, with N fixed, controls the Metropolis ratio of DIU. Let g be a genotype at L loci, that is, g is sequence of allele pairs of length L. The allele frequencies at each loci, $\gamma^{(l)}$ for $l = 1, \ldots, L$, depend on the number of alleles at the locus. Indeed, as long the number of alleles at a single locus increase, the values of γ in that locus become smaller. For instance, a locus with six alleles has $(6 \cdot 7)/2 = 21$ possible genotypes. If the allele frequencies are assigned equally probable for all of them, then the values of γ in that locus are all equal to $1/21$, which is a small probability. As the number of alleles in a locus is large, the values in γ become smaller.

Further, the number of loci also have a significant effect on g. Because of the independence among loci, $f(g|\gamma) = \gamma^{(1)} \ldots \gamma^{(L)}$. The combination of large number of alleles (i.e. small values of γ's), and large number of loci causes small values of $f(g|\gamma)$. In Eq. (5.7), for the first case in which n does not change for the introduction of a genotype already present in the list of unique genotypes, $r \approx 1$. In this case the proposal that keeps the value of n unchanged is accepted with high probability. For the second case of n increasing, r tends to zero, which implies that the proposal attempting to increase the value of n is rejected with high probability. For the third case of n decreasing, r grows to infinity, thus the proposal for decreasing the value of n is accepted with high

probability.

The acceptance rate of the DIU sampler is affected, then, by $L \to \infty$ and $\gamma \to 0$. That is, if the number of loci and the number of alleles are large, the proposals of g, in which the n has increased, are rejected with probability close to 1.0. In other words, at each iteration of the DIU algorithm, n is reduced in one unit or unaltered with high probability. The probability of increasing n is very low under these conditions. Figure 6.3 showed this behaviour for DIU.

Because the DIU algorithm is impractical with a large number of alleles, it may be applied when diallelic locus are involved. In addition, the DIU algorithm may be implemented in situations where n does not change, or at least, it lies in a minimal range of values. The toy example in Section 6.1 supports the validity of the chain generated by the DIU algorithm, as it achieved to simulate the posterior distribution.

A solution for the drawback exposed above should consider the effect of L and γ in the proposal distribution of the DIU sampler. A possible solution may be conceiving the proposal distribution as a mixture of two distributions. Suppose the jth row of g is chosen for updating. With probability π, a candidate value is drawn from a normalised version of the proposal distribution in Eq. (5.4), and with probability $1-\pi$ is sampled over a set of genotypes which are co-referent with the jth genotype. This distribution will depend on p only, because it is defined in terms of $f(g^{\text{obs}}|g,p)$. Then, L, m, γ are not involved. Perhaps a difficulty with this approach is the normalising constants, but it is expected that these sums can be efficiently computed. This brief explanation opens the door to possible solutions to fix DIU, as it is an idea to pursue in future research.

7.3 Computational comparison

This section aims to compare the algorithms regarding the number of operations required in a single iteration. As explained in Section 3.2.2, GENUAD is a Gibbs algorithm for updating $\mathcal{G}|X$ and $X|\mathcal{G}$, where $\mathcal{G}$ is a $N \times L$ matrix with the true genotypes in the entire population, and X is a $N \times S$ indicator matrix. From Section 5.1.1, SMERED$^+$ is an RJMCMC algorithm for jointly updating G and y, where G is a $n \times L$ matrix with the true genotypes of those individuals that were observed in the sample and y is a vector of S indices.

Following Section 3.2.2.1, GENUAD updates $\mathcal{G}|X$ row by row (individual by individual), and for a specific row, locus by locus. The aim here is counting the number of operations required for computing $\lambda_{il}^{(k)}$ defined in Eq. (3.3) for the individual i at locus l with $k = 1, \ldots, \eta_l$. The first factor

$$\prod_{j \in \mathcal{I}} \prod_{r=1}^{R} \Pr(g_{jlr}^{\text{obs}} | \mathcal{G}_{il}^{(k)}, p)$$

requires $|\mathcal{I}|R$ operations, where $|\mathcal{I}|$ is the number of samples in which the individual i appeared, and R is the number of PCR replicates. So, for a fixed value of k, calculating $\lambda_{il}^{(k)}$ requires $|\mathcal{I}|R+2$ operations. For updating the ith row of $\mathcal{G}$, $(|\mathcal{I}|R+2)\sum_{j=1}^{L}\eta_L$ operations are required. Since $|\mathcal{I}|\leq S$, at the most, $(SR+2)\sum_{j=1}^{L}\eta_L$ operations are required for updating a row of $\mathcal{G}$. Because $\mathcal{G}$ has N rows, the total number of operations required by GENUAD for updating $\mathcal{G}$ given X is $O_1 = N(SR+2)\sum_{j=1}^{L}\eta_L$, at the most. If the individual i does not appear in the sample, the number of operations is smaller, as Eq. (3.4) suggests.

From Section 3.2.2.2, $X|\mathcal{G}$ is updated column by column. For the jth column of X, the term λ_{ji} defined in Eq. (3.6) requires $L+2$ operations, for a fixed value of i, where $i = 1, \ldots N$. Thus, GENUAD requires $O_2 = SN(L+2)$ operations for updating X given $\mathcal{G}$.

Therefore, for updating the pair $(\mathcal{G}, X)$, an iteration of GENUAD requires at the most $O_1 + O_2$ operations, where O_1 and O_2 are given as above.

Recall that SMERED$^+$ starts by choosing a pair of observations in g^{obs}, and splitting or merging them, depending on the corresponding values in y. The case iii) in Section 5.1.1.1 provides the upper bound for the number of operations needed for SMERED$^+$. For one of the alleles, the support is $1, 2, \ldots, m_l$ where m_l is the number of alleles at locus l. For generating a new genotype (when splitting or merging), there are $\prod_{j=1}^{L}(m_j - 1)$ possible genotypes (Example 5.1 illustrates this point). Some of them may have probability zero depending on the set of observations related with the pair chosen. For each of the possible genotypes, two factors are computed: $\Pr(g|\gamma)$ and $\Pr(g^{\text{obs}}|g, p)$. Each requires the computation of L operations, plus the product of all of them. So, a possible genotype requires $2L+1$ operations. Thus, SMERED$^+$ requires $(2L+1)\prod_{j=1}^{L}(m_j - 1)$ at the most.

Note that the number of operations required by SMERED$^+$ does not depend on the observed sample size S. It depends on the number of loci and the number of alleles at each locus. This is an advantage of SMERED$^+$ respect to GENUAD. Not only S does influence the maximum number of operations required by GENUAD but also the population size, the number of loci, the number of PCR replicates, and the number of alleles at each locus. Therefore, SMERED$^+$ should not struggle with larger datasets.

For the specific applications in Chapter 6, for which $S = 47$ and the number of iterations was 200 000, the GENUAD algorithm takes approximately 55 minutes, while SMERED$^+$ takes around 15 minutes. Clearly, this information benefits SMERED$^+$, which has shown that mixes faster than GENUAD in the case of $R > 2$. As seen, large values of R will increase the number of operations of GENUAD and, so, the computational time.

Additionally, it is well known that the single nucleotide polymorphisms (SNPs) involves a larger number of loci, with few alleles, than the microsatellite markers.

From the discussion above, the number of operations required for both the GENUAD and the SMERED$^+$ algorithms increase as the number of loci L increase. It suggests that the use of SNPs rather than microsatellite markers demands more computational resources for these algorithms. Besides, the previous section discussed the effect of a large value of L on the DIU algorithm. When L is large, the proposals for which the value of n increases are rejected with high probability. Thus, there is not advantage in the use of SNPs when implementing the DIU algorithm. In general, the use of SNPs would require a more extensive model to incorporate inheritance and linkage. This is a topic that needs further exploration.

Chapter 8

Conclusion and future work

The general problem addressed in this thesis is that there is an uncertainty associated with the correct assignment of genotypes to individuals, and this uncertainty is part of the model that endeavours to uncover their unique and true identity. In consequence, the number of unique observations in the collected sample is unknown. Wright et al. (2009) and Steorts et al. (2016) proposed two different Bayesian models for incorporating this uncertainty. The simulations were carried out via MCMC algorithms, namely GENUAD and SMERED, respectively.

This thesis determined the convergence of the Markov chains produced by these algorithms. In the case of GENUAD, once the irreducibility of the chain was determined other convergence properties followed, such as ergodicity. Because the positivity condition does not hold for the specific support, irreducibility was concluded by constructing a sequence of states holding the condition in Besag's lemma. This result also guarantees the capability of the full conditional densities to supply the joint density of interest. Thus, the existence and uniqueness of the invariant distribution is ensured. On the other hand, a counterexample showed that the chain in SMERED does not converge to the desired posterior distribution for which the chain was constructed. This failure was attributed to an oversight related to the dimension change in one of the parameters.

The detailed study of these algorithms allowed to discover and address some inaccuracies and imprecisions. For example, the controversial combinatorial term for updating X given $\mathcal{G}$ in Wright et al. was examined and clarified. Additionally, the fact that Steorts et al. considered a symmetric proposal distribution indicated the presence of convergence issues. Thus, a comprehensive understanding of the aforementioned models and algorithms led to the development of a new approach for solving the same misidentification problem, the SMERED$^+$ algorithm. It is a trans-dimensional algorithm which considers the dimension change of the parameter space, and it is an improved version of SMERED. Another algorithm was proposed, named the DIU algorithm. It is a Metropolised independent sampler which requires a minimal computational effort. It worked well for the small-scale example, but unfortunately, it failed when simulating extensive datasets. The positivity of the proposal distributions ensured the convergence to the invariant distribution of these two new approaches.

The GENUAD and SMERED$^+$ algorithms were compared via simulations. The posterior distribution of interest was taken from Wright et al.. Both GENUAD and SMERED$^+$ reach convergence, but differently. The degree of corruption in the data played a decisive role in the efficiency of the algorithms. In the specific data considered in Wright et al., more corruption in the data implies a low correlation between the parameters. If the data has high levels of contamination, then the parameters would be almost independent because there would be many possibilities to move around the state space. The fact that GENUAD is a Gibbs algorithm may limit its performance if the parameters are highly correlated. Since SMERED$^+$ is an RJMCMC algorithm for jointly updating the parameters, it is impervious to correlations between them.

When using MCMC methods for sampling from a target distribution, the real challenge is assessing convergence, mainly because there is no consensus about how to do it. Instead, an extensive list of diagnostics attempts to determine, at least approximately, if the sample has been taken from the same distribution. However, no singular diagnostic will be definitive regarding the convergence of the simulated sample. Thus, simulations via MCMC algorithms and the study of their convergence is a controversial matter. For instance, debated issues regarding variations in the styles of simulating include the pertinence of constructing a single long chain or two chains, the benefits of discarding the initial simulations, and the practicality of thinning the chains. All of these subjects lack definitive answers. Thus, there is an element of subjectivity in the analysis of Markov chains.

Individually, GENUAD and SMERED$^+$ accomplished Markov chains which sample from the same distribution, as shown by the examples. Reviewing different strategies for assessing convergence of MCMC algorithms, Cowles and Carlin (1996) stated that "multiple algorithms may also be helpful, because each will have its own convergence properties and may reveal different features of the likelihood or posterior surface". Thus, this is a case where two different algorithms, GENUAD and SMERED$^+$, are used for sampling from the same posterior distribution, each with its own convergence properties. The weaknesses of one algorithm can be the strengths of the other. For example, with the information provided by the GENUAD algorithm in the case of $R \geq 2$, there is no conclusive way to know if there is a failure to sample a representative portion of the state space of interest. As seen, GENUAD struggles to leave subregions of that space. Nevertheless, SMERED$^+$ showed more freedom to explore the entire state space.

While the results suggest that GENUAD and SMERED$^+$ are promising algorithms tailored to situations that involve misreported data or some kind of measurement error, a limitation is that the model considered here is problem-specific. The algorithms have not been applied to different datasets rather than that in Wright et al. (2009). However, they could apply to frequently used datasets in economics, finance, public policy analysis, and population research. Some examples are mentioned below.

For instance, the Integrated Data Infrastructure (IDI) is an extensive research database containing confidential longitudinal microdata about people, households, and business in New Zealand. Data is from a range of government agencies, Statistics NZ

surveys including the 2013 Census, and non-government organisations. The flow of data in the IDI includes linking the identities across the different sources using deterministic and probabilistic linking. The algorithms proposed here may be useful in this clearing data step. The Ministry of Social Development's Child Youth & Family Data and the 2013 New Zealand Census are examples of datasets in the IDI that provide the raw information analysed in numerous studies. However, they are known to have considerable data quality issues as they include partial observations and misreported data.

The models considered here may also be useful in economic applications. Financial data can have misreported values when there is a substantial price movement on a trade that is later declared by the stock exchange to have been erroneous. Many critical macroeconomic variables are published at sub-annual frequencies and then revised at a later date across Organisation for Economic Co-operation and Development (OECD) countries[1]. A different, but related, use of the algorithms proposed in this thesis is associated with self-reporting surveys. Berg and Lien (2006) proposed a statistical model that simultaneously controls for misreporting and survey non-response. The model assumes that misreporting and non-response events are jointly distributed as a multinomial logit. Because there is evidence that the sexual orientation may be connected to economic variables such as personal income, household income, geographical location and health outcomes, Berg and Lien (2009) further analysed survey misreporting using an online survey that includes self-reported rates of lying. The approach proposed here could be used alongside that in Berg and Lien to estimate probabilities of misreporting and non-response based on a clean dataset, and thus, provide more precise estimates of the size of the non-heterosexual population.

The above instances and countless other survey datasets are examples where the novel technique proposed and analysed in this thesis could be applied. They allow accountability for measurement error that researchers struggle to control, specially as these datasets are used as the basis for thousands of peer-reviewed articles.

While possible applications were described above, the previous chapter discussed various subjects that promise future research. For example, the laziness tendency of SMERED$^+$ may be used to solve periodicity and mixing problems, and the overall proposal distribution of the DIU model could be improved by constructing a new proposal as a mixture of densities. In addition, a possible generalisation of the solution to the misidentification problem could be in the context of *fuzzy cluster analysis*.

Cluster analysis consists in partitioning a data set into a number of subsets such that the members of a cluster have a specific degree of similarity. This problem of data clustering "has been widely studied in data mining and machine learning literature because of its numerous applications to summarization, learning, segmentation, and target marketing", according to Aggarwal (2013, p. 2).

Vazirgiannis et al. (2003) define *crisp clustering* as the clustering undertaken where

[1] http://www.oecd.org/sdd/40315408.pdf

there is discrete 0 or 1 membership of objects to a cluster. That is, an element in the data either belongs to a class or not. This type of clustering assumes the existence of strictly defined boundaries between the clusters. Nevertheless, this assumption is not always valid since such boundaries can be fuzzy. As the author stated (p. 143), "a more detailed description of an object's membership in a cluster is needed since there are cases that will assign each object to more than one cluster with a different degree of belief". Thus, *fuzzy clustering* may be considered a more effective approach when accounting for the uncertainty included in data.

In general, for fuzzy cluster analysis techniques the membership is not a dichotomous variable, instead, it defines a degree of membership between 0 and 1 for each element assigned into a cluster. Thus, each object could belong to more than one cluster with a different degree of belief depending on its similarity with other objects in the clusters. The aim is to find groups of similar objects, comprising the clusters, in a set of S given by $\{g_1, \ldots, g_S\}$. The degree of belief with which g_j, for $j = 1, \ldots, S$, belong to a cluster i is arranged in a $n \times S$ matrix denoted by U, where n is the number of clusters. Because the fuzzy partitioning allows membership at any of the n clusters, then U could be any object of the set of n-partitions. That is, $U \in \mathcal{M}_{n,S}$ where $\mathcal{M}_{n,S}$ denotes the space of all $n \times S$ matrices such that $u_{ij} \in [0, 1]$ for $j = 1, \ldots, S$ and $j = 1, \ldots, S$; $\sum_{i=1}^{n} u_{ij} = 1$ for $i = 1, \ldots, n$; and $\sum_{j=1}^{S} u_{ij} > 0$ for $j = 1, \ldots, S$.

The problem with this approach is that the number of clusters is unknown, that is, the number of rows of the matrices U varies. However, the connection with the clustering area is evident and promises future research.

Appendices

Appendix A

Definitions for GENUAD

A.1 Compatible genotypes

As previously mentioned, the primary theme of this thesis is not founded on genetics. This appendix introduces some definitions to make the data accessible and treatable. The definitions required to understand the applications have been modified using a different language but keeping consistency with the biological definition.

Definition A.1.1. Let $A \subset \mathbb{Z}^+$ the set of possible alleles at a single locus. Then the *genotype* of an individual at that locus is a pair (x, y) where $x, y \in A$. There is no notion of order in this definition, that is, (x, y) and (y, x) refer to the same genotype.

Definition A.1.2. Let be (x, y) a genotype at a single locus. If $x = y$ then the genotype is *homozygous*. Otherwise, it is *heterozygous*.

Definition A.1.3. Let $A \subset \mathbb{Z}^+$ the set of possible alleles at a single locus with $|A| = m$ (i.e. m alleles at the locus). Under allelic dropout, if (x, x) with $x \in A$ is an observed genotype, then the set of possible true genotypes is given by $C = \{(x, y) : \text{for all } y \in A\}$. These genotypes will be called *compatible genotypes*.

For example, consider a single locus with three alleles, say $A = \{1, 2, 3\}$. According to Definition A.1.3, if the observed genotype is $(3, 3)$ then the compatible genotypes are $\{(1, 3), (2, 3), (3, 3)\}$. If the observed genotype is $(1, 3)$, then $(1, 3)$ is its unique compatible genotype. Table A.1 shows the compatible genotypes for all other cases.

Table A.1: Compatibility between true and observed genotypes.

		True					
		(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(3,3)
Observed	(1,1)	✓	✓	✓			
	(1,2)		✓				
	(1,3)			✓			
	(2,2)		✓		✓	✓	
	(2,3)					✓	
	(3,3)			✓		✓	✓

Definition A.1.4. A set of observed genotypes is *co-referent* if they have a common compatible genotype.

This definition is a borrowed concept from the record linkage terminology. Notice that it establishes a relation only between observed genotypes. Table A.2 pairs for co-referent genotypes.

Table A.2: Co-reference between observed genotypes.

		Observed					
		(1,1)	(1,2)	(1,3)	(2,2)	(2,3)	(3,3)
Observed	(1,1)	✓	✓	✓	✓		✓
	(1,2)	✓	✓		✓		
	(1,3)	✓		✓			✓
	(2,2)	✓	✓		✓	✓	✓
	(2,3)				✓	✓	✓
	(3,3)	✓		✓	✓	✓	✓

Notice that the co-reference relation is not transitive. That is, if s_1 and s_2 are co-referent samples, and also s_2 and s_3 are co-referent, then, s_1 and s_3 could be not co-referent. For instance, $(1,1)$ and $(2,2)$ are co-referent because they could be associated to the true genotype $(1,2)$; and $(2,2)$ and $(2,3)$ to $(2,3)$. However, $(1,1)$ and $(2,3)$ are not co-referent, because does not exist a common latent genotype that can be associated to them.

A.2 The corruption process

A.2.1 Allelic dropout in GENUAD

The corruption in the data considered in Wright et al. (2009) is due to the presence of allelic dropout. The definition of compatibility between genotypes, given in Definition A.1.3, limits the set of possible true genotypes for one observed. In general, if the alleles in a single locus are labelled with capital letters, then a genotype could be homozygous (AA) or heterozygous (AB). The conditional density $f(g^{\text{obs}}|g,p)$ is given by

If $g = \text{AA}$ then

$$\Pr(g^{\text{obs}}|g,p) = \begin{cases} 1 & \text{for } g^{\text{obs}} = \text{AA}, \\ 0 & \text{for } g^{\text{obs}} = \text{AB}. \end{cases}$$

If $g = \text{AB}$ then

$$\Pr(g^{\text{obs}}|g,p) = \begin{cases} p/2 & \text{for } g^{\text{obs}} = \text{AA or BB}, \\ 1-p & \text{for } g^{\text{obs}} = \text{AB}. \end{cases}$$

Notice that true homozygous genotypes are free of genotyping error. For true heterozygous, there are three possibilities. If AB is the true genotype:

- It may be wrongly observed as AA, that is, a failure to detect the allele B, with probability $p/2$.
- It may be wrongly observed as BB, that is, a failure to detect the allele A, with probability $p/2$.
- It may be correctly observed as AB with probability $1 - p$.

Figure A.1 illustrates these conditional probabilities.

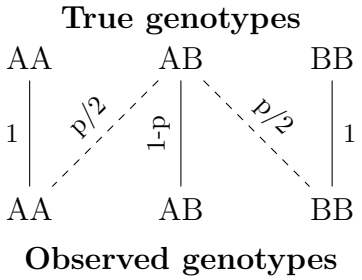


Figure A.1: Conditional probabilities of the observed genotypes given the true genotype. Pr(Observed|True).

Appendix A. Definitions for GENUAD

Bibliography

Aggarwal, C. C. (2013). An introduction to cluster analysis. In Aggarwal, C. C. and Reddy, C. K., editors, *Data clustering: Algorithms and applications*, chapter 1, pages 1–27. Chapman and Hall/CRC, Boca Raton, FL. 129

Arnold, B. C. and Press, S. J. (1989). Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156. 24

Athreya, K. and Lahiri, S. (2006). *Measure Theory and Probability Theory*. Springer, New York. 15, 16, 18, 22

Barker, R. J. and Link, W. A. (2013). Bayesian multimodel inference by RJMCMC: A Gibbs sampling approach. *The American Statistician*, 67(3):150–156. 30

Barker, R. J., Schofield, M. R., Wright, J. A., Frantz, A. C., and Stevens, C. (2014). Closed-population capture-recapture modeling of samples drawn one at a time. *Biometrics*, 70(4):775–782. 6, 40, 48, 49

Basu, R., Hermon, J., and Peres, Y. (2017). Characterization of cutoff for reversible Markov chains. *The Annals of Probabability*, 45(3):1448–1487. 119

Behrends, E. (2000). *Introduction to Markov chains: With special emphasis on rapid mixing*. Vieweg, Braunschweig. 92

Berg, N. and Lien, D. (2006). Same-sex sexual behaviour: US frequency estimates from survey data with simultaneous misreporting and non-response. *Applied Economics*, 38(7):757–769. 129

Berg, N. and Lien, D. (2009). Sexual orientation and self-reported lying. *Review of Economics of the Household*, 7(1):83–104. 129

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236. 24

Besag, J. (1994). Discussion: Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1734–1741. 25

Biffi, D. and Williams, D. A. (2017). Use of non-invasive techniques to determine population size of the marine otter in two regions of Perú. *Mammalian Biology*, 84:12–19. 6

Brémaud, P. (1999). *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. Springer, New York. 15, 18, 19, 23, 92

Bromaghin, J. F. (2007). The genetic mark-recapture likelihood function of capwire. *Molecular Ecology*, 16(23):4883–4884. 48

Brooks, S. and Roberts, G. (1998). Convergence assessment techniques for Markov chain Monte Carlo. *Statistics and Computing*, 8(4):319–335. 33, 106

Brooks, S. P. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455. 34

Brooks, S. P., Giudici, P., and Roberts, G. O. (2003). Efficient construction of reversible jump markov chain monte carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):3–39. 85

Casella, G. and George, E. I. (1992). Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174. 23

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335. 27, 28, 29, 70

Chiu, R., Akolekar, R., Zheng, Y., Leung, T., Sun, H., Chan, K., Lun, F., Go, A. T., Lau, E., To, W., Leung, W., Tang, R., Au-Yeung, S., Lam, H., Kung, Y., Zhang, X., van Vugt, J., Minekawa, R., Tang, M., Wang, J., Oudejans, C., Lau, T. K., Nicolaides, K. H., and Lo, Y. (2011). Non-invasive prenatal assessment of trisomy 21 by multiplexed maternal plasma DNA sequencing: Large scale validity study. *British Medical Journal*, 342. 5

Christen, P. (2012a). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, Berlin. 12, 55

Christen, P. (2012b). A survey of indexing techniques for scalable record linkage and deduplication. *Knowledge and Data Engineering, IEEE Transactions on*, 24(9):1537–1555. 12

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904. 33, 128

Cox, D. R. and Miller, H. D. (1965). *The Theory of Stochastic Processes*. Methuen, London. 15

Ditmer, M., Vincent, J., Werden, L., Tanner, J., Laske, T., Iaizzo, P. A., Garshelis, D., and Fieberg, J. (2015). Bears show a physiological but limited behavioral response to unmanned aerial vehicles. *Current Biology*, 25(17):2278–2283. 5

Farr, W. M., Mandel, I., and Stevens, D. (2015). An efficient interpolation technique for jump proposals in reversible-jump Markov chain Monte Carlo calculations. *Royal Society open science*, 2(6). 85

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210. 12

Forshew, T., Murtaza, M., Parkinson, C., Gale, D., Tsui, D., Kaper, F., Dawson, S.-J., Piskorz, A., Jimenez-Linan, M., Bentley, D., Hadfield, J., May, A., Caldas, C., Brenton, J., and Rosenfeld, N. (2012). Non-invasive identification and monitoring of cancer mutations by targeted deep sequencing of plasma DNA. *Science Translational Medicine*, 4(136):136–168. 5

Frantz, A. C., Pope, L. C., Carpenter, P. J., Roper, T. J., Wilson, G. J., Delahay, R. J., and Burke, T. (2003). Reliable microsatellite genotyping of the Eurasian badger (Meles meles) using faecal DNA. *Molecular Ecology*, 12(6):1649–1661. 8, 9, 107

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman & Hall/CRC, Boca Raton, FL. 22, 27, 28, 36, 74

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472. 33

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-6(6):721–741. 22

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bernardo, J., Berger, J., Dawid, A., and Smith, J., editors, *Bayesian Statistics 4*, pages 169–193. Oxford University Press, Oxford. 34

Green, P. (2003). Trans-dimensional Markov chain Monte Carlo. In Green, P., Hjort, N. L., and Richardson, S., editors, *Highly structured stochastic systems*, chapter 6, pages 179–198. Oxford University Press, Oxford. 30

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732. 29, 76, 77

Green, P. J. and Richardson, S. (2001). Modelling heterogeneity with and without the Dirichlet process. *Scandinavian Journal of Statistics*, 28(2):355–375. 75

Hammersley, J. M. and Clifford, P. (1971). Markov fields on finite graphs and lattices. Unpublished. 24

Hastie, D. I. and Green, P. J. (2012). Model choice using reversible jump Markov chain Monte Carlo. *Statistica Neerlandica*, 66(3):309–338. 30, 31, 32, 85

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109. 32

Heidelberger, P. and Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operational Research*, 31(6):1109–1144. 35

Herzog, T. N., Scheuren, F. J., and Winkler, W. E. (2007). *Data Quality and Record Linkage Techniques*. Springer, New York, 1st edition. 12, 13

Hobert, J. and Casella, G. (1998). Functional compatibility, Markov chains, and Gibbs sampling with improper posteriors. *Journal of Computational and Graphical Statistics*, 7(1):42–60. 24

Hobert, J., Robert, C., and Goutis, C. (1997). Connectedness conditions for the convergence of the Gibbs sampler. *Statistics and Probability Letters*, 33(3):235–240. 24, 25, 69

Jain, S. and Neal, R. M. (2004). A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182. 75

Jerrum, M. (2003). *Counting, Sampling and Integrating: Algorithms and Complexity*. Springer-Verlag, Berlin. 119

Lange, K. (1999). *Numerical Analysis for Statisticians*. Springer, New York. 22

Lange, K. (2002). *Mathematical and Statistical Methods for Genetic Analysis*. Springer, New York. 22

Levin, D. A., Peres, Y., and Wilmer, E. L. (2009). *Markov chains and mixing times*. American Mathematical Society. 15, 92, 119

Levine, R. A. and Casella, G. (2006). Optimizing random scan Gibbs samplers. *Journal of Multivariate Analysis*, 97(10):2071–2100. 23

Lin, S., Thompson, E., and Wijsman, E. (1993). Achieving irreducibility of the Markov chain Monte Carlo method applied to pedigree data. *Mathematical Medicine and Biology*, 10(1):1–17. 22

Link, W. A. and Barker, R. J. (2010). *Bayesian Inference with Ecological Applications*. Academic Press, London. 22

Liu, J. S. (1996). Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119. 15, 32, 33, 91

Liu, J. S. (2008). *Monte Carlo strategies in scientific computing*. Springer, New York. 20, 33, 36, 91

Lukacs, P. M. and Burnham, K. P. (2005a). Estimating population size from DNA-based closed capture–recapture data incorporating genotyping error. *The Journal of Wildlife Management*, 69(1):396–403. 7

Lukacs, P. M. and Burnham, K. P. (2005b). Review of capture–recapture methods applicable to non-invasive genetic sampling. *Molecular Ecology*, 14(13):3909–3919. 6, 7

Maletic, J. and Marcus, A. (2010). Data cleansing: A prelude to knowledge discovery. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook*, chapter 2, pages 19–30. Springer. 12

Marucco, F., Vucetich, L. M., Peterson, R. O., Adams, J. R., and Vucetich, J. A. (2012). Evaluating the efficacy of non-invasive genetic methods and estimating wolf survival during a ten-year period. *Conservation Genetics*, 13(6):1611–1622. 6

Meyn, S. P. and Tweedie, R. (1993). *Markov chains and stochastic stability*. Springer-Verlag, New York. 15, 19

Miller, C. R., Joyce, P., and Waits, L. P. (2005). A new method for estimating the size of small populations from genetic mark-recapture data. *Molecular Ecology*, 14(7):1991–2005. 48

Mondol, S., Ullas Karanth, K., Samba Kumar, N., Gopalaswamy, A. M., Andheria, A., and Ramakrishnan, U. (2009). Evaluation of non-invasive genetic sampling methods for estimating tiger population size. *Biological Conservation*, 142(10):2350–2360. 6

Morin, D. J., Kelly, M. J., and Waits, L. P. (2016). Monitoring coyote population dynamics with fecal DNA and spatial capture-recapture. *Journal of Wildlife Management*, 80(5):824–836. 6

Newcombe, H. B., Kennedy, J. M., Axford, S. J., and James, A. P. (1959). Automatic linkage of vital records. *Science*, 130(3381):954–959. 12

Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978). Statistical inference from capture data on closed animal populations. *Wildlife Monographs*, (62):3–135. 5, 6, 7

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11. 33

Raftery, A. and Lewis, S. (1992). How many iterations in the Gibbs sampler? In Bernardo, J., Berger, J., Dawid, A., and Smith, J., editors, *Bayesian Statistics 4*, pages 763–773. Oxford University Press, Oxford. 35

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792. 76

Robert, C. P. and Casella, G. (2004). *Monte Carlo statistical methods*. Springer, New York. 15, 17, 19, 20, 22, 23, 25, 26, 27, 29, 36, 70, 92

Roques, S., Furtado, M., Jácomo, A. T. A., Silveira, L., Sollmann, R., Tôrres, N. M., Godoy, J. A., and Palomares, F. (2014). Monitoring jaguar populations panthera onca with non-invasive genetics: A pilot study in Brazilian ecosystems. *Oryx*, 48(3):361–369. 6

Sheehan, N. and Thomas, A. (1993). On the irreducibility of a Markov chain defined on a space of genotype configurations by a sampling scheme. *Biometrics*, 49(1):163–175. 22

Sisson, S. A. (2005). Transdimensional Markov chains. *Journal of the American Statistical Association*, 100(471):1077–1089. 30

Sorensen, D. and Gianola, D. (2002). *Likelihood, Bayesian and MCMC methods in quantitative genetics*. Springer-Verlag, Berlin. 21

Steorts, R. C., Hall, R., and Fienberg, S. E. (2016). A Bayesian approach to graphical record linkage and deduplication. *Journal of the American Statistical Association*, 111(516):1660–1672. 3, 4, 12, 13, 39, 40, 50, 51, 52, 53, 54, 56, 57, 58, 59, 70, 77, 78, 127

Thompson, E. A. (2000). Monte Carlo methods on genetic structures. In Barndorff-Nielsen, O. E., Cox, D. R., and Klüppelberg, C., editors, *Complex Stochastic Systems*, number 87 in Monographs on Statistics and Applied Probability, chapter 4, pages 175–233. Chapman and Hall/CRC, Boca Raton, FL. 22

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics*, pages 1701–1728. 22, 25

Torra, V. (2010). Privacy in data mining. In Maimon, O. and Rokach, L., editors, *Data mining and knowledge discovery handbook*, chapter 35, pages 687–716. Springer, New York. 13

Vazirgiannis, M., Halkidi, M., and Gunopulos, D. (2003). *Uncertainty handling and quality assessment in data mining*. Springer-Verlag, London. 129

Waagepetersen, R. and Sorensen, D. (2001). A tutorial on reversible jump MCMC with a view toward applications in QTL-mapping. *International Statistical Review*, 69(1):49–61. 29

Wright, J. A. (2011). *Incorporating Genotype Uncertainty into mark-recapture-Type models for Estimating Abundance using DNA Samples*. PhD thesis, University of Otago. 4, 8, 116

Wright, J. A., Barker, R. J., Schofield, M. R., Frantz, A. C., Byrom, A. E., and Gleeson, D. M. (2009). Incorporating genotype uncertainty into mark-recapture-type models for estimating abundance using DNA samples. *Biometrics*, 65(3):833–840. 3, 4, 6, 7, 10, 12, 13, 39, 40, 41, 42, 43, 44, 46, 47, 48, 49, 53, 54, 57, 59, 62, 66, 70, 73, 77, 79, 83, 84, 92, 93, 98, 99, 107, 127, 128, 134

Yoshizaki, J., Brownie, C., Pollock, K., and Link, W. (2011). Modeling misidentification errors that result from use of genetic tags in capture-recapture studies. *Environmental and Ecology Statistics*, 18(1):27–55. 7